Tallinna Ülikool

Kvantitatiivne digihumanitaaria

Jaagup Kippar

2019

Sisukord

Sissejuhatus

11. klassi koolimatemaatikas tutvutakse põhiliste statistilise andmeanalüüsi mõistete ja arvutuskäikudega ning õpitakse neid kättesaadavate andmete peal kasutama. Sealsed miinimum, maksimum, mediaan ning aritmeetiline keskmine koos standardhälbega on arvutuste aluseks. Nende abiga saab andmetest esialgse ja põhilise ülevaate.

Kooliajast tuttavatele arvutustele lisaks leidub andmete analüüsimiseks ka hulgaliselt muid meetodeid mis - kord üks kord teine - kasulikuks osutuvad. Neid proovides ja kombineerides saab vaadata, et kas andmetest ka midagi sellist välja paistab, mis esimese hooga kohe silma ei jää. Materjali tagumisest poolest leiab mitu meetodit, kus uuritavate tunnuste arvu vähendatakse - nii et algselt mõõdetud mitmest või mitmeteistkümnest jääb vaid mõni alles - ning siis on suhteid ja rühmi juba kergem joonisele kuvada ja sealt seoseid otsida.

Omamoodi harjumist nõuab mõtteviis, kus "jah" või "ei" vastuse asemel öeldakse "95% tõenäosusega võime väita, et" - ja järgneb vastuse eeldatav vahemik. Kui tahetakse väite õiguses kindlam olla, siis tuleb vahemikku laiendada, kui lepitakse sagedasemate valeotsustega, siis saab ka uskuda vastuse suuremasse täpsusesse - paratamatult tuleb ühelt poolt võites mujal järele anda.

Suuremas osas raamatus kasutatakse programmeerimiskeelt nimega R - mis on 2010ndatel taas laialt levima hakanud. Lõpuosas tehakse märgatav osa arvutustest ka Pythoni ja tema lisapakettide abil läbi - nii on pärast meetodite toimimisest aru saamist võimalik valida kahe levinuma andmetöötluskeele seast, kummaga parajasti põhjust lähemalt tegemist teha.

Näited võetakse enamasti keelevaldkonnast - loetakse tekstides sõnu, täishäälkuid ja muid tunnuseid, mis suurelt jaolt lapsepõlvest tuttavad. Eks oma uuringute juures asendab igaüks andmed omale tarvilikega. Arvutab tulemused välja ning siis loodetavasti enne suuremat välja kuulutamist mõtleb enne vähemalt korra, et mida leitud erinevused mingis valdkonnas tähendavad ning kui suure õigsustõenäosuse puhul võib järgmisi järeldusi seni leitutele tuginema panna. Mõndapidi kipub ju ebakindel tunduma, et vastus pole kindel "jah" või "ei". Samas kui arvutuskäik näitab, et vale vastuse tõenäosus on üks miljoni või miljardi kohta, siis see on palju kindlam vastus, kui mõni lihtsalt üle huulte või ekraani tulnud jah-sõna.

Materjalile eelneb digihumanitaaria tehnilisemaid vahendeid üldisemalt tutvustav konspekt "Digihumanitaaria tehnoloogiad" - kui mõni siinne järeldus või arvutus tundub liialt äkiline olema, siis sealt võib leida pikemalt seletavaid ning teisest suunast tulevaid näiteid.

Tutvus R-iga

“R on mõnes mõttes kohutav, aga midagi paremat pole ka välja mõeldud” ütles üks meditsiinistatistikaga tegelev tuttav. Siinses materjalis kasutame selle keele abi mitmesuguste teemade seletamisel.

Käivitada on R-i käske alustuseks lihtsam ühekaupa käsurealt. Hiljem saab ka pikemaid programmilõike kokku panna ning neid eraldi tööle lükata.

Kui Linuxi alla R installitud, siis piisab käivitamiseks üldiselt üherealisest käsust

```
R
```

Mujal tuleb paigaldada eraldi R keskkond soovi korral koos värvilisema R-studioga. Sõltumata ettevalmistuskohast, on lõpuks ikkagi silme ees käsuviip

```
>
```

Arvutused

Anname käske ette, tema annab vastused vastu. Kirjutame 3+2, saame vastuseks 5. Kantsulgudes üks vastuse ees näitab, et tegemist on esimese vastusega. Kuna andmeanalüüsi juures võib andmeid ja vastuseid palju olla, siis tuleb nummerdamine kasuks.

```
> 3+2
[1] 5
```

Muud tavalised tehtemärgid -, * ja / on lahutamise, korrutamise ja jagamise kohta. Astendamiseks sobib kaks järjestikust korrutusmärki

```
> 2**5
[1] 32
```

ruutjuure võtmiseks käsklus sqrt.

```
> sqrt(25)
[1] 5
```

Andmete meelde jätmiseks muutujad.

Soovituslikuks omistuskäsuks R-keeles on noole ehk siis “väiksem kui” ja miinusmärgi kombinatsiooni abil omistamine. Muutuja väärtust saab näha lihtsalt selle nime trükkides

```
> pikkus <- 168
> pikkus
[1] 168
```

Nool võib olla ka teises suunas

```
> 169 -> pikkus
> pikkus
[1] 169
```

Töötab ka mitmest muust keelest tuttav võrdusmärgiga omistamine

```
> pikkus=165
> pikkus/100
[1] 1.65
```

Harjutus

- Korruta kaks arvu
- Leia, mitu protsenti eesti rahvastikust moodustavad 26000 Eesti Filmi Andmebaasis märgitud isikut.
- Suurim arv rolle filmi kohta selles andmebaasis on 296. Kui palju oleks sama osakaal inimesi eesti rahvastikust

```
> 3*5
[1] 15
> 26000/1320000
[1] 0.01969697
> (26000/1320000)*100
[1] 1.969697
> round((26000/1320000)*100, 2)
[1] 1.97
> paste(round((26000/1320000)*100, 2), '%')
[1] "1.97 %"
> 296/26000
[1] 0.01138462
> 296/26000*1320000
[1] 15027.69
```

Andmekogum

Lihtsamaks neist vektor ehk kollektsioon, kus sama tüüpi väärtused reas on. R-i kasulik eripära, et kogumiga võib tehteid teha peaaegu sama vabalt kui üksikute väärtustega. Lihtsamaks üksikute väärtuste ritta kokku kogumise käsuks on c - nagu collection

```
> pikkused <- c(165, 172, 180)
```

Liidan väärtustele ühe ja näengi tulemust

```
> pikkused+1
[1] 166 173 181
```

Sama sentimeetriteks arvutamise juures

```
> pikkused/100
[1] 1.65 1.72 1.80
```

Lubatakse liita ka terve sama pikk andmevektor korraga

```
> pikkused <- c(165, 172, 180)
> kontsad <- c(1, 5, 2)
> pikkused + kontsad
[1] 166 177 182
```

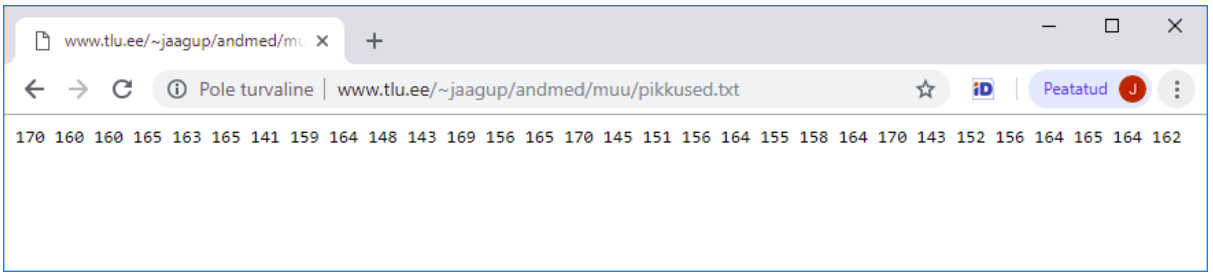
Suurem kogus andmeid on mugav sisse lugeda failist - näiteks veebis asuvast failist.

Aadressilt

```
http://www.tlu.ee/~jaagup/andmed/muu/pikkused.txt
```

leiab arvud

```
170 160 160 165 163 165 141 159 164 148 143 169 156 165 170 145 151 156 164 155 158 164 170
143 152 156 164 165 164 162
```



Arvurea sisse lugemiseks sobib käsklus scan

```
> pikkused <- scan("http://www.tlu.ee/~jaagup/andmed/muu/pikkused.txt")
Read 30 items
```

Edasi tavalised arvujada arvutused.

Vähim:

```
> min(pikkused)
[1] 141
```

Suurim:

```
> max(pikkused)
[1] 170
```

Aritmeetiline keskmine

```
> mean(pikkused)
[1] 158.9
```

Mediaan:

```
> median(pikkused)
[1] 161
```

Vahemik vähimast suurimani:

```
> range(pikkused)
[1] 141 170
```

Kokkuvõte andmetest: vähim, esimene veerand (millest 25% arvudest väiksemad), mediaan, aritmeetiline keskmine, kolmas veerand, maksimum.

```
> summary(pikkused)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  141.0   155.2   161.0   158.9   164.8   170.0
```

Arvude kättesaamine kogumist.

Kõigepealt arvude kogus

```
> length(pikkused)
[1] 30
```

ja arvud ise

```
> pikkused
 [1] 170 160 160 165 163 165 141 159 164 148 143 169 156
[14] 165 170 145 151 156 164 155 158 164 170 143 152 156
[27] 164 165 164 162
```

Andmete algusots - ülevaade suuremast kogumist

```
> head(pikkused)
[1] 170 160 160 165 163 165
```

Esimene element. Tähelepanuks, et erinevalt näiteks Pythonist hakkab siin lugemine ühest

```
> pikkused[1]
[1] 170
```

Teine element

```
> pikkused[2]
[1] 160
```

Loetelus viimased

```
> tail(pikkused)
[1] 152 156 164 165 164 162
```

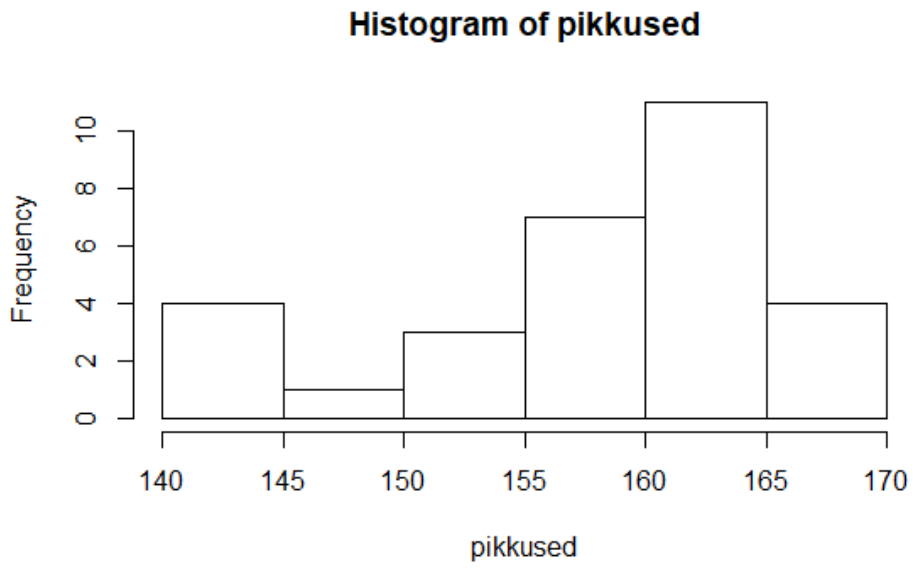
Viimane element ehk element, mille järjekorranumbriks on elementide koguarv

```
> pikkused[length(pikkused)]
[1] 162
```

Histogramm

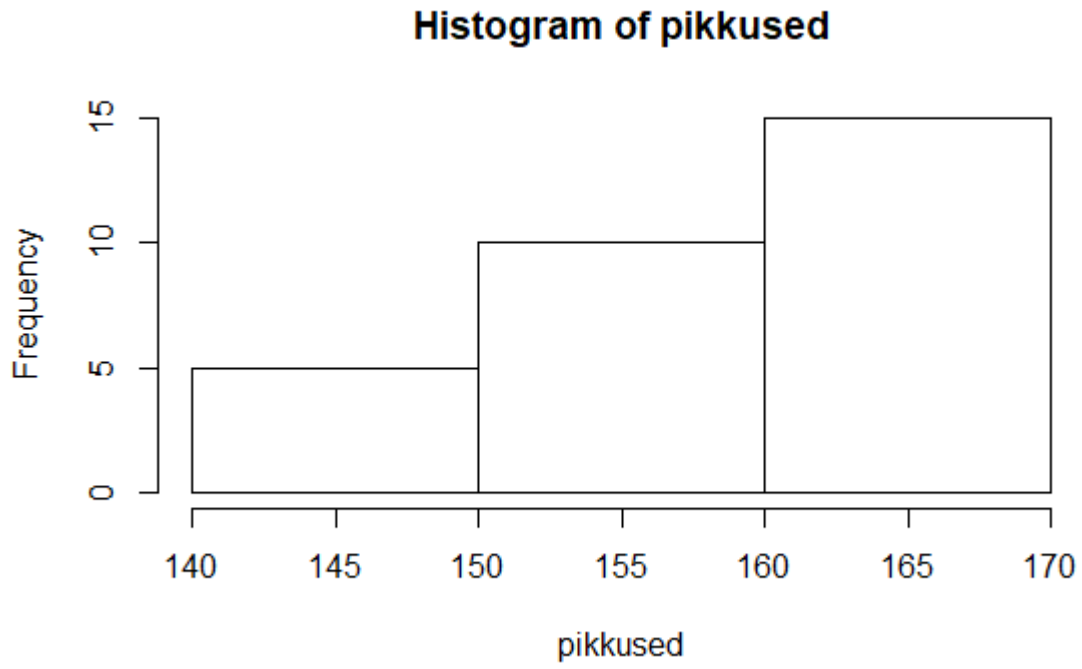
Tundmatu arvukogumi uurimiseks soovitatakse teha kõigepealt ülevaateks histogramm.

```
> hist(pikkused)
```



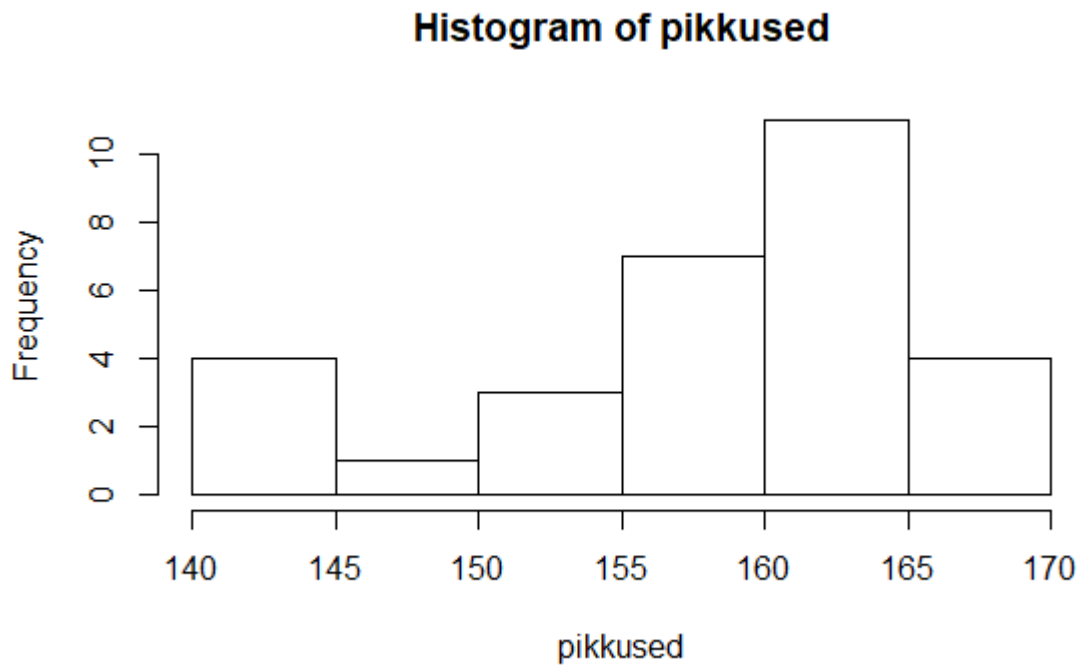
Tegemist kõrvuti tulpadega oleva diagrammiga, kus iga vahemiku puhul näha, et mitu arvu sellesse satub. Nii tekib silme ette ülevaade arvude jaotusest. Joonise koostamisele saab aga ka vihjeid anda, kui soovida midagi rõhutada, pehmendada või lihtsalt mugavamalt vaadatavaks teha. Saab anda soovitusliku katkestuskohtade arvu

```
> hist(pikkused, breaks=3)
```



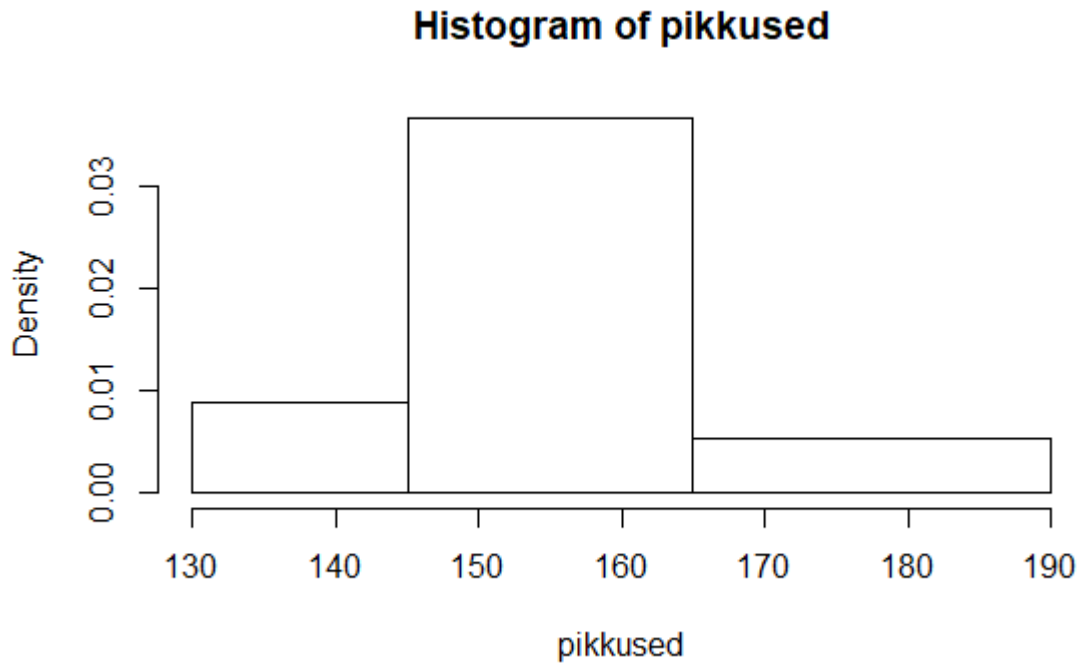
R aga katsub leida tasakaalu soovitu ning mõistliku vahel ning kümne küsitud koha asemel tuleb praeguse arvukoguse juures vaid kuus tulpa.

```
> hist(pikkused, breaks=10)
```



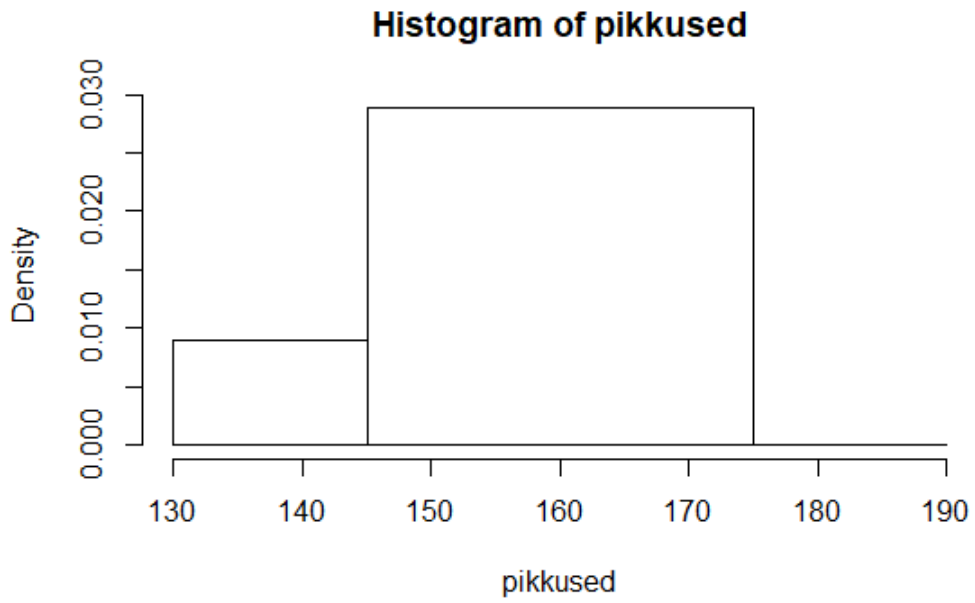
Võib ka määrata sentimeetrid, kus tulbad osadeks jaotatakse. Eripikkuste vahemike puhul näidatakse aga mitte üldarvu, vaid suhtelist tihedust, nii et pindala järgi paistab, kui ühtlaselt on tulemused vahemikku “laiali määritud”

```
> hist(pikkused, breaks=c(130, 145, 165, 190))
```



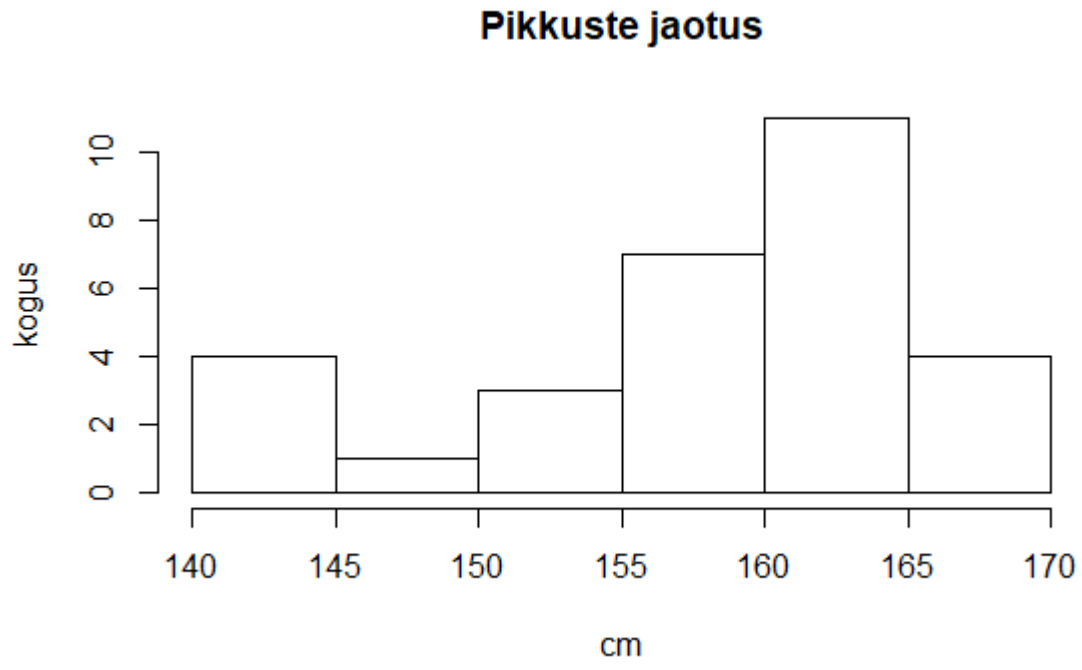
Nihutades viimast vahet veidi edasi, muutub aga praeguse väikese arvukoguse juures pilt märgatavalt

```
> hist(pikkused, breaks=c(130, 145, 175, 190))
```



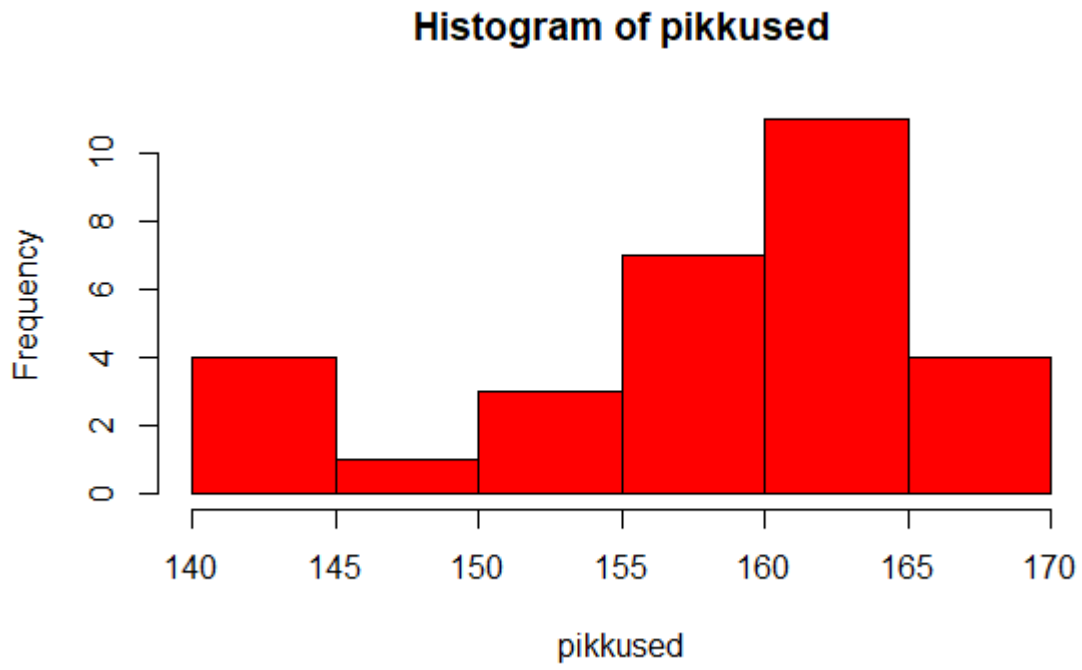
Joonistele on viisakas lisada pealkiri ning telgede kirjeldused

```
> hist(pikkused, main = "Pikkuste jaotus", xlab="cm", ylab="kogus")
```



Värvimisnäide

```
> hist(pikkused, col="red")
```

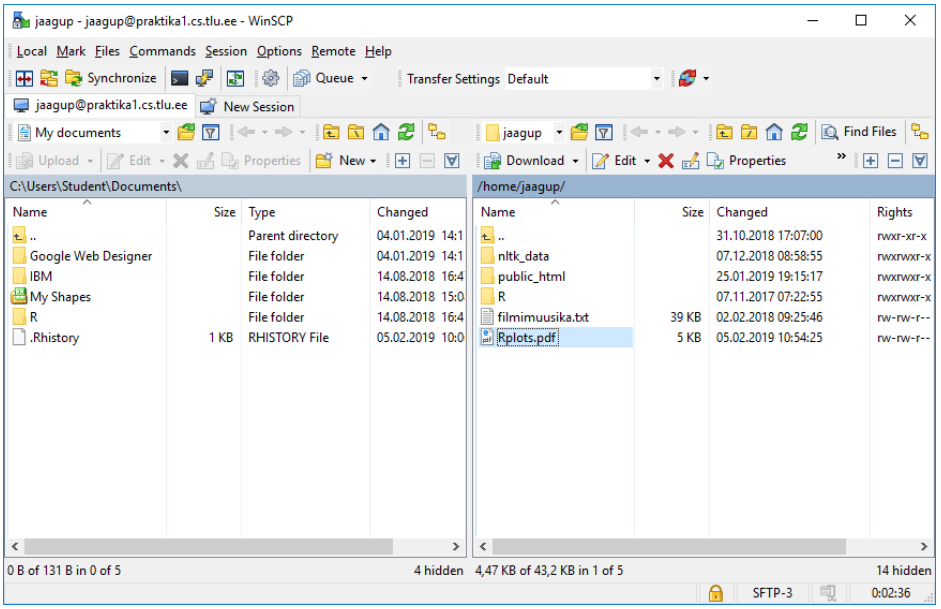


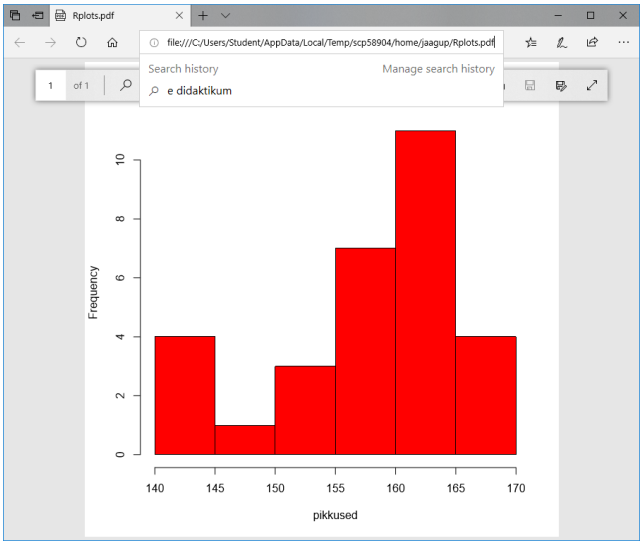
Kui jooniseid koostada Linuxi käsureal näiteks putty terminaliaknas, siis tuleb joonise nägemiseks pärast selle koostamist kirjutada käsklus dev.off()

Tulemusena salvestatakse joonis aktiivses kataloogis olevasse faili nimege Rplots.pdf

Faili saab avada kas koha peal või siis kopeerides kohalikku masinasse ja seal uurides.

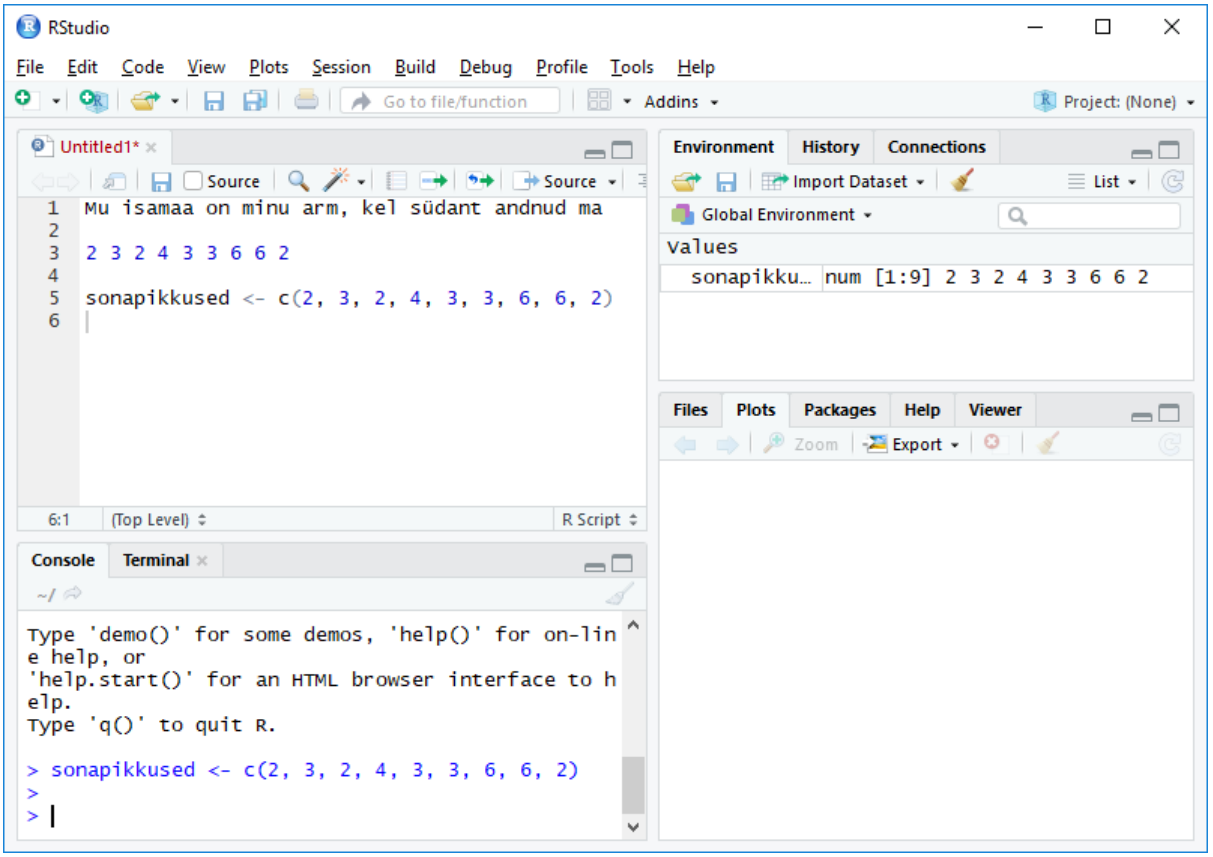
```
> pikkused<-scan("http://www.tlu.ee/~jaagup/andmed/muu/pikkused.txt")
Read 30 items
> hist(pikkused, col="red")
> dev.off()
null device
          1
```





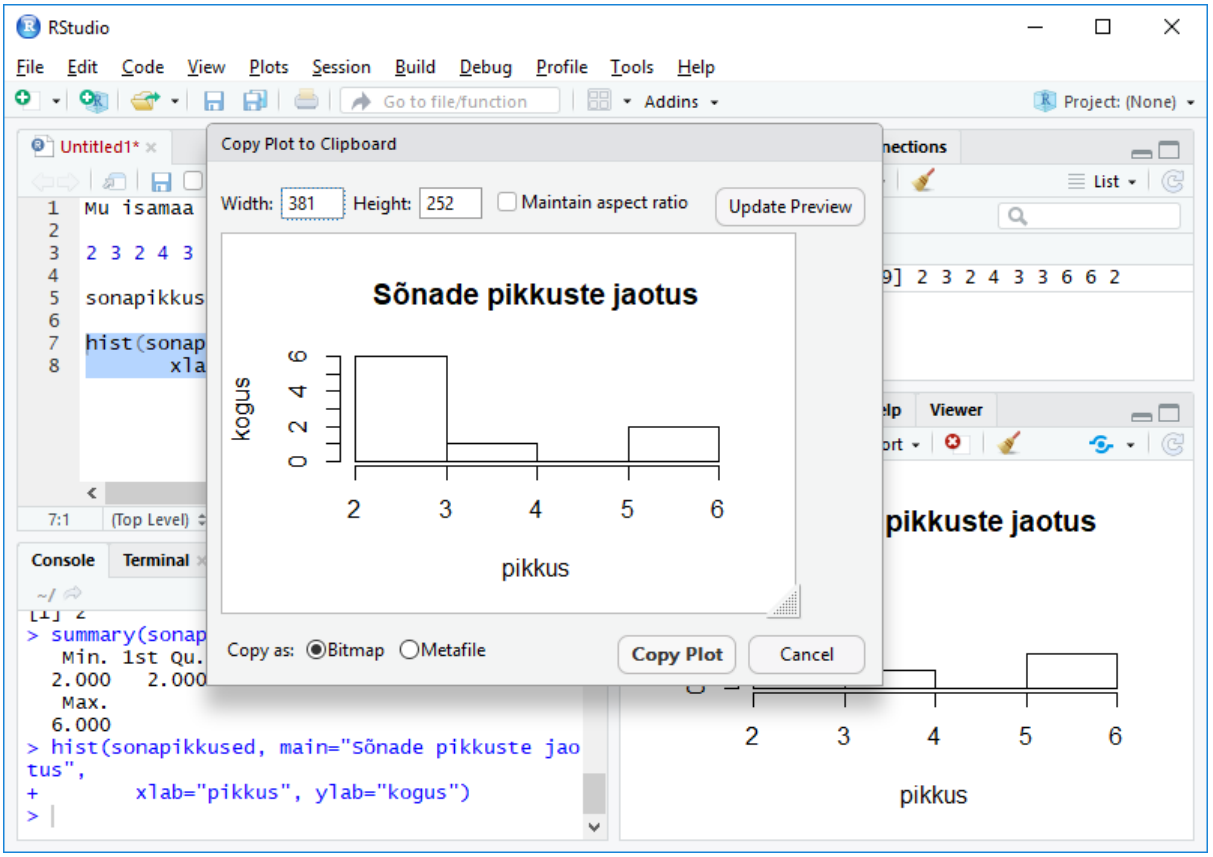
Harjutus

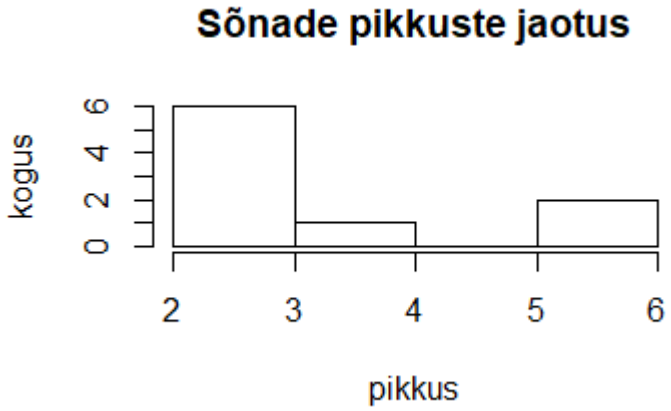
- Otsi paari lausega tekst, koosta käsitsi sõnade tähtede arvudest arvujada, omista R-i muutujale.
- Kuva vähim ja suurim väärtus, mediaan ja aritmeetiline keskmine
- Joonista sõnapikkuste histogramm, lisa joonise pealkiri ja telgede kirjeldused
- Joonista sarnane sõnapikkuste histogramm, kasutades andmeid failist
  `http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_sonapikkused.txt`




```
> min(sonapikkused)
[1] 2
> summary(sonapikkused)
   Min. 1st Qu.  Median    Mean 3rd Qu.
  2.000   2.000   3.000   3.444   4.000
   Max.
  6.000
>
```

```
> hist(sonapikkused, main="Sõnade pikkuste jaotus",
       xlab="pikkus", ylab="kogus")
```





Filtreerimine, järjestamine

Kogumist saab andmed küsida välja vastavalt tingimusele. Näiteks 160 sentimeetrist lühemad pikkused

```
> pikkused[pikkused<160]
 [1] 141 159 148 143 156 145 151 156 155 158 143 152 156
```

Madalama piiri puhul tuleb ka pikkusi vähem

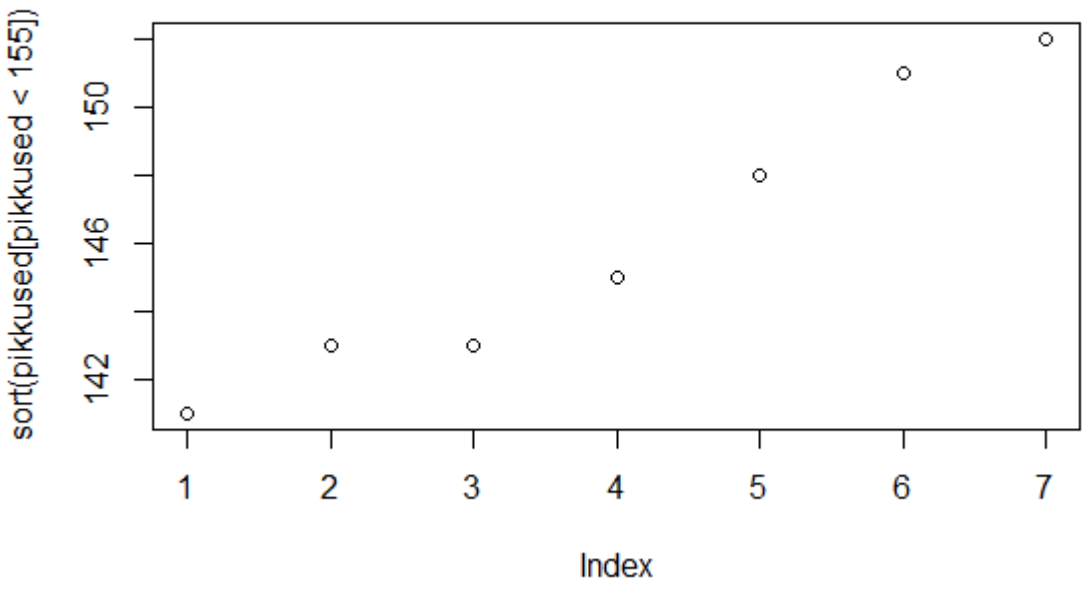
```
> pikkused[pikkused<155]
[1] 141 148 143 145 151 143 152
```

Järjestamiseks sobib käsklus sort

```
> sort(pikkused[pikkused<155])
[1] 141 143 143 145 148 151 152
```

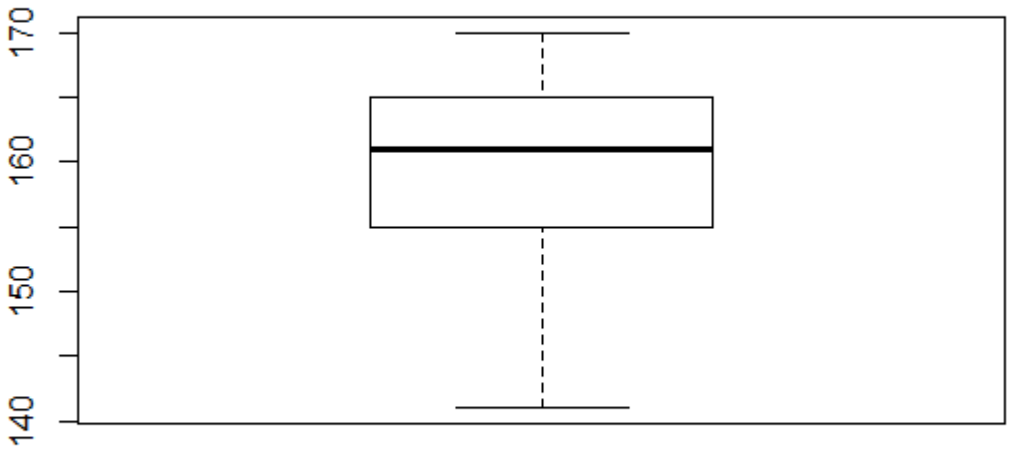
Järjestatud pikkused täppidena joonisele. Joonise x-teljeks võetakse järjekorranumber ehk indeks ning y-teljeks väärtus

```
> plot(sort(pikkused[pikkused<155]))
```



Nagu histogrammi, nii ka karpdiagrammi peetakse heaks mooduseks arvukogumist ülevaate andmisel. Keskmine rasvane joon on mediaan. Keskel oleva karbi alumine külg alumine kvartiil - millest on väiksemaid väärtusi veerand - ning ülemine külg ülemine kvartiil. Karbi küljes olevad vurrud näitavad maad vähima ja suurima väärtuseni.

```
> boxplot(pikkused)
```



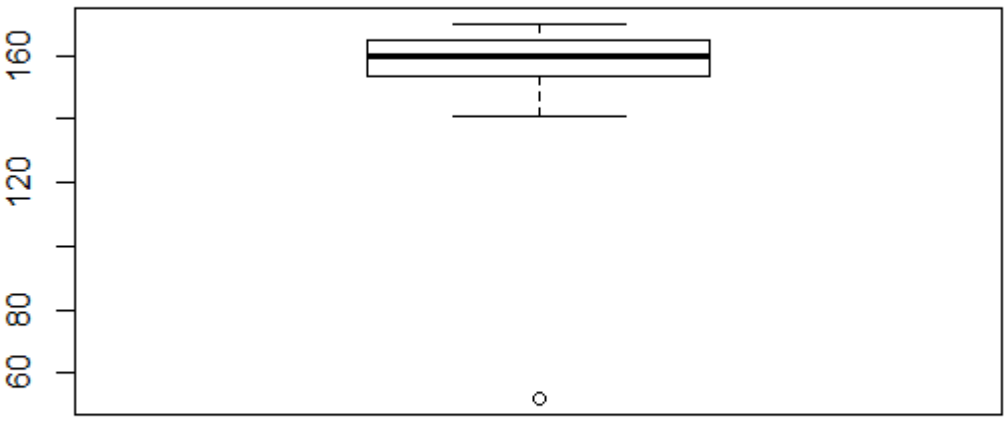
Kogumisse väärtuse lisamiseks tehakse uus nõnda, et sinna jäetakse vana sisu + juurde veel vajalik element.

```
> pikkused <- c(pikkused, 52)
> pikkused
 [1] 170 160 160 165 163 165 141 159 164 148 143 169 156 165 170
[16] 145 151 156 164 155 158 164 170 143 152 156 164 165 164 162
[31]  52
```

Nii paistab, et see vastsündinud lapse pikkus ka loetelus juures.

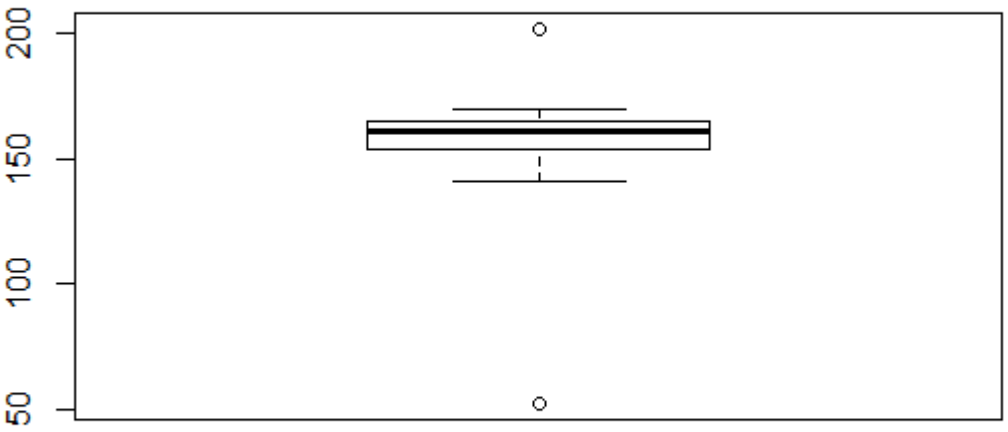
Kui mõni väärtus on teistest väga erinev, siis boxplot ei paiguta seda üldise karpdiagrammi hulka, vaid näitab punktina eraldi.

```
> boxplot(pikkused)
```



Eelnevas näites muutsime pikkuste kogumit. Siin kuvamise juures lisame joonisele juurde küll ühe 202-sentimeetri pikkuse korvpalluri, aga ei paiguta teda pikkuste kogumisse.

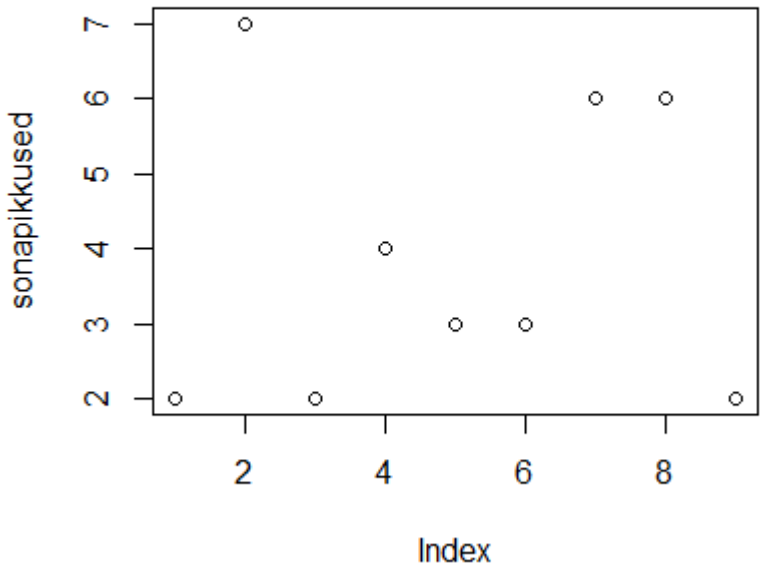
```
> boxplot(c(pikkused, 202))
```



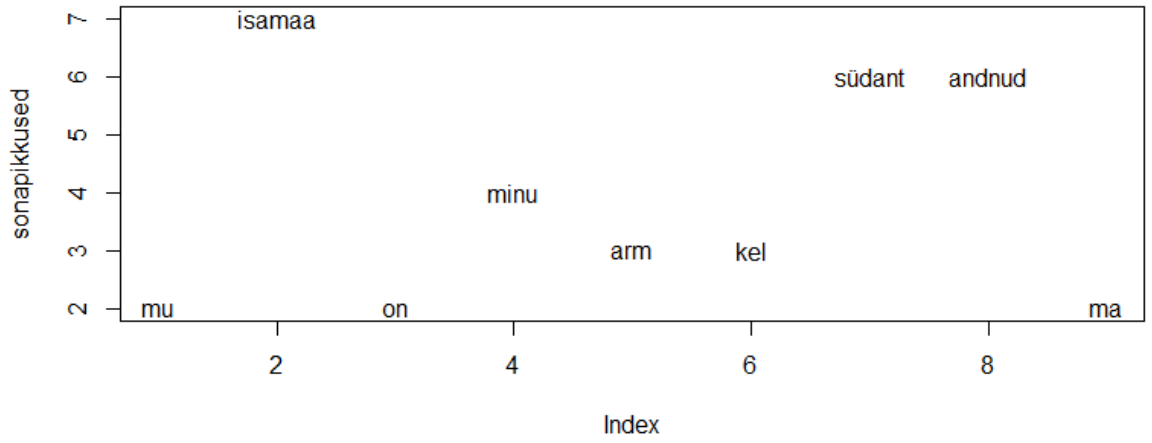
Harjutus

- Koosta/otsi loetelu lause(te) sõnapikkustega.
- Kuva sõnapikkused täppidena ekraanile, sõna järjekorranumber on indeksiks
- Järjesta pikkused ja kuva taas sama joonis
- Koosta sõnapikkustest karpdiagramm

```
sonad=c("mu", "isamaa", "on", "minu", "arm", "kel", "südant", "andnud", "ma")
sonapikkused=c(2, 7, 2, 4, 3, 3, 6, 6, 2)
plot(sonapikkused)
```




```
plot(sonapikkused, type="n")
text(1:length(sonapikkused), sonapikkused, sonad)
```



Tidyverse

R-i “kohutavat” süsteemitust on püütud parandada mitmete lisateekidega. Neist populaarsemaks osutunud ning omavahel enamvähem ühilduvad on kogutud teeki nimega tidyverse.

Teegid ehk lisapaketid on üldsegi programmeerimiskeelte juures päris tähtsad. Uue teema alustamisel ja keele/vahendite valikul võib keele mugavusest või tuntusest määravamaks osutuda vajalike pakettide olemasolu. Näiteks geneetikas on R valitseval kohal justnimelt valdkonnas arendatud paljude tarvilike teekide tõttu.

Esimesel korral kasutamiseks tuleb teek installida (kui seda automaatselt juba sees ei ole). Käskluseks siinsel juhul

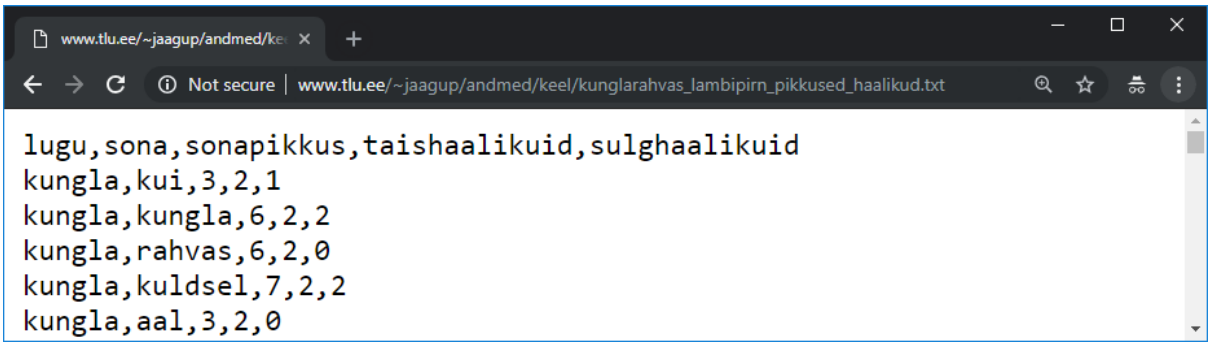
```
> install.packages("tidyverse")
```

Tulemusena ragistab masin mõne kuni mõnikümmend minutit ning õiguste ja kettamahu sobivuse korral sikutab paketi kohale. Edasi see on juba olemas ning järgmistel kordadel piisab kasutamiseks vaid teegi sisse lugemisest.

```
> library(tidyverse)
Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr
Conflicts with tidy packages ----------------------------
filter(): dplyr, stats
```

```
lag():    dplyr, stats
```

Andmeid on mugav sisse lugeda csv-failist



Selleks tidyverse puhul sobib käsklus read_csv

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")

Parsed with column specification:
cols(
  lugu = col_character(),
  sona = col_character(),
  sonapikkus = col_integer(),
  taishaalikuid = col_integer(),
  sulghaalikuid = col_integer()
)
```

Andmete algusotsast annab ülevaate head

```
> head(sonad)
# A tibble: 6 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
4 kungla kuldsel          7             2             2
5 kungla aal              3             2             0
6 kungla kord             4             1             2
```

lõpuotsast tail

```
> tail(sonad)
# A tibble: 6 x 5
  lugu      sona   sonapikkus taishaalikuid sulghaalikuid
  <chr>     <chr>       <int>         <int>         <int>
1 lambipirn pea             3             2             1
2 lambipirn kuklas          6             2             2
3 lambipirn ja              2             1             0
4 lambipirn suu             3             2             0
5 lambipirn pärani          6             3             1
6 lambipirn lahti           5             2             1
```

juhuslikest kohtadest ülevaade käsuga sample_n, praegusel juhul kokku kümme rida

```
> sample_n(sonad, 10)
# A tibble: 10 x 5
   lugu      sona       sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>           <int>         <int>         <int>
 1 lambipirn siin                4             2             0
 2 lambipirn millesti            8             3             1
 3 lambipirn enne                4             2             0
 4 kungla    hääl                4             2             0
 5 lambipirn istuvad             7             3             2
 6 lambipirn jne                 3             1             0
 7 kungla    laulis              6             3             0
 8 lambipirn jne                 3             1             0
 9 lambipirn matemaatik         10             5             3
10 kungla    vanemuine           9             5             0
```

Dollarimärgi abil saab tabeli tulba sõnad kätte eelnevalt tuttava vektorina

```
> sonad$sona
  [1] "kui"                 "kungla"
  [3] "rahvas"              "kuldsel"
  [5] "aal"                 "kord"
  [7] "istus"               "maha"
  [9] "sööma"               "siis"
 [11] "vanemuine"           "murumaal"
```

Sealt võimalik järjekorranumbri järgi vastus küsida. Sõna "kungla" on laulus teisel kohal. Kantsulgudes üks näitab, et tegemist on esimese (ja praegusel juhul ainukese) vastusega.

```
> sonad$sona[2]
[1] "kungla"
```

Laulu alguse sõnade kätte saamiseks samuti käsklus head

```
> head(sonad$sona, 10)
 [1] "kui"     "kungla"  "rahvas"  "kuldsel" "aal"     "kord"
 [7] "istus"   "maha"    "sööma"   "siis"
```

Tulpadeks olevatele andmevektoritele saab rahus rakendada eelnevalt tuttavaid funktsioone. Siin näiteks leitakse suurim sõnapikkus.

```
> max(sonad$sonapikkus)
[1] 19
```

Erinevad väärtused toob välja käsk unique

```
> unique(sonad$lugu)
[1] "kungla"    "lambipirn"
```

Päringud

Andmetega mängimine ehk data wrangling on R-i üks tähtis osa. Järjestamiseks käsklus `arrange`

```
> arrange(sonad, sonapikkus)
# A tibble: 672 x 5
   lugu      sona  sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>      <int>         <int>         <int>
 1 kungla    ja             2             1             0
 2 kungla    ja             2             1             0
 3 kungla    ja             2             1             0
```

Kahanevasse järjekorda panekuks saab järjestada negatiivse sõnapikkuse järgi

```
> arrange(sonad, -sonapikkus)
# A tibble: 672 x 5
   lugu      sona                sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>                    <int>         <int>         <int>
 1 lambipirn politseijaoskonnale         19             9             3
 2 lambipirn intelligentselt             15             5             4
 3 lambipirn funktsioneeriva             15             7             2
 4 lambipirn märkimisväärset             15             6             2
 5 lambipirn valgusallikata              14             6             3
```

tekstiliste andmete puhul tuleb aga abiliseks `desc`

```
> arrange(sonad, desc(sona))
# A tibble: 672 x 5
   lugu      sona          sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>              <int>         <int>         <int>
 1 lambipirn üllatuslikult         13             5             3
 2 lambipirn üles                   4             2             0
 3 lambipirn üldjoontes            10             4             2
 4 lambipirn üks                    3             1             1
```

Filtreerimiseks ehk sobivate alles jätmiseks käsklus `filter`

```
> filter(sonad, lugu=="kungla")
# A tibble: 75 x 5
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid
   <chr>  <chr>        <int>         <int>         <int>
 1 kungla kui              3             2             1
 2 kungla kungla           6             2             2
 3 kungla rahvas           6             2             0
 4 kungla kuldsel          7             2             2
 5 kungla aal              3             2             0
```

Käske võib sobivalt üksteise sisse paigutada. Kõigepealt jäetakse alles Kungla rahva sõnad ning siis järjestatakse kahanevalt sõnapikkuse järgi

```
> arrange(filter(sonad, lugu=="kungla"), desc(sonapikkus))
# A tibble: 75 x 5
   lugu   sona      sonapikkus taishaalikuid sulghaalikuid
```

```
   <chr>  <chr>          <int>         <int>         <int>
 1 kungla vanemuine          9             5             0
 2 kungla laululugu          9             5             1
 3 kungla lauluviis          9             5             0
 4 kungla vanemuise          9             5             0
 5 kungla murumaal           8             4             0
 6 kungla murueide           8             5             1
 7 kungla kuldsel            7             2             2
 8 kungla mängima            7             3             1
```

Pikemate päringute puhul saab aga üksteise sees olevaid sulge nõndamoodi palju ning paremaks peetakse süntaksikuju, kus andmed igast käsust järgmisesse `%>%` operaatori abil edasi suunatakse

```
> sonad %>% filter(lugu=="kungla")
# A tibble: 75 x 5
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid
   <chr>  <chr>        <int>         <int>         <int>
 1 kungla kui              3             2             1
 2 kungla kungla           6             2             2
 3 kungla rahvas           6             2             0
 4 kungla kuldsel          7             2             2
 5 kungla aal              3             2             0
```

Sealt vajaduse korral omakorda edasi, nii et pikemates päringutes võib sarnaseid suunamisi üle kümne olla

```
> sonad %>% filter(lugu=="kungla") %>% arrange(desc(sonapikkus))
# A tibble: 75 x 5
   lugu   sona      sonapikkus taishaalikuid sulghaalikuid
   <chr>  <chr>          <int>         <int>         <int>
 1 kungla vanemuine          9             5             0
 2 kungla laululugu          9             5             1
 3 kungla lauluviis          9             5             0
 4 kungla vanemuise          9             5             0
 5 kungla murumaal           8             4             0
 6 kungla murueide           8             5             1
 7 kungla kuldsel            7             2             2
 8 kungla mängima            7             3             1
```

Nii saab käsku vajadusel ka mitmele reale jagada, suunamisoperaator aga peab jääma rea lõppu.

```
sonad %>%
   filter(lugu=="kungla") %>%
   arrange(desc(sonapikkus))
```

Käsureaaknas näidatakse senikaua rea algul plussmärke, kui arvatakse, et käsklus läheb veel edasi

```
> sonad %>%
+     filter(lugu=="kungla") %>%
+     arrange(desc(sonapikkus))
```

```
# A tibble: 75 x 5
   lugu   sona      sonapikkus taishaalikuid sulghaalikuid
   <chr>  <chr>          <int>         <int>         <int>
 1 kungla vanemuine          9             5             0
 2 kungla laululugu          9             5             1
 3 kungla lauluviis          9             5             0
 4 kungla vanemuise          9             5             0
 5 kungla murumaal           8             4             0
 6 kungla murueide           8             5             1
 7 kungla kuldsel            7             2             2
 8 kungla mängima            7             3             1
```

Suuremas päringus on nõnda võimalik soovi korral etappe kommenteerimise abil ka ajutiselt eraldada - praegu järjestatakse kõikide tabelis olevate laulude sõnad

```
sonad %>%
   #filter(lugu=="kungla") %>%
   arrange(desc(sonapikkus))
```

```
> sonad %>%
+    #filter(lugu=="kungla") %>%
+    arrange(desc(sonapikkus))
# A tibble: 672 x 5
   lugu      sona                sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>                    <int>         <int>         <int>
 1 lambipirn politseijaoskonnale         19             9             3
 2 lambipirn intelligentselt             15             5             4
 3 lambipirn funktsioneeriva             15             7             2
 4 lambipirn märkimisväärset             15             6             2
 5 lambipirn valgusallikata              14             6             3
 6 lambipirn topsivendadelt              14             5             5
 7 lambipirn naerukrampides              14             6             3
 8 lambipirn reflektoorsete              14             6             3
 9 lambipirn kodumaalastel               13             6             3
10 lambipirn elektrituruga               13             6             4
# ... with 662 more rows
```

Harjutus

- Tehke näited läbi
- Leidke kolme sulghäälikuga sõnad
- Järjestage kolme sulghäälikuga sõnad sõnapikkuse järgi kahanevalt
- Leidke suurim sulghäälikute arv sõnade tabelis

Grupeerimine

Tabelis lihtsaimaks kokkuvõtete arvutamiseks sobib käsklus `summarise`. Parameetrina tuleb ette lugeda, mida arvutada soovitakse ning kuidas uued tulbad nimetatakse

```
> sonad %>% summarise(sonadearv=n(), aritmkeskmine=mean(sonapikkus),
mediaan=median(sonapikkus))
# A tibble: 1 x 3
  sonadearv aritmkeskmine mediaan
      <int>         <dbl>   <dbl>
1       672          5.76       5
```

Soovides tulemusi rühmitada ühe tunnuse erinevate väärtuste - näiteks laulunime kaupa, siis tuleb vahele paigutada group_by

```
sonad %>% group_by(lugu) %>%
  summarise(sonadearv=n(), aritmkeskmine=mean(sonapikkus), mediaan=median(sonapikkus))

# A tibble: 2 x 4
  lugu      sonadearv aritmkeskmine mediaan
  <chr>         <int>         <dbl>   <int>
1 kungla           75          4.76       4
2 lambipirn       597          5.89       5
```

Nii näeb ridade kaupa tulemusi kummagi laulu kohta eraldi. Tegemist levinud võimalusega, kus saab vabalt valida, millise tunnuse järgi andmestikku rühmadeks jaotada ning milliseid käsklusi nendele andmetele rakendada.

Harjutus

- Tehke näited läbi
- Arvutage aritmeetiline keskmine ja mediaan täishäälikute kohta kummagi laulu sõnades
- Koosta laulude sõnade põhjal tabel, kus igal real on erinev täishäälikute arv sõnas, tulpadeks on suurim ning vähim sulghäälikute arv vastavates sõnades
- Koosta Kungla rahva laulu sõnade järgi tabel, kus igal real on sulghäälikute arv sõnas ning veergudeks on täishäälikute arvu keskmine ja mediaan

Lahendusi

Täishäälikute andmed laulude kaupa

```
sonad %>% group_by(lugu) %>%
  summarise(taish_keskm=mean(taishaalikuid), taish_mediaan=median(taishaalikuid))

# A tibble: 2 x 3
  lugu      taish_keskm taish_mediaan
  <chr>           <dbl>         <int>
1 kungla           2.27             2
2 lambipirn        2.67             2
```

Sulghäälikute arv täishäälikute arvu kaupa

```
sonad %>% group_by(taishaalikuid) %>%
  summarise(min_sulg=min(sulghaalikuid), max_sulg=max(sulghaalikuid))
```

```
# A tibble: 9 x 3
  taishaalikuid min_sulg max_sulg
          <int>    <dbl>    <dbl>
1             0        0        0
2             1        0        2
3             2        0        3
4             3        0        4
5             4        0        5
6             5        0        5
7             6        1        4
8             7        2        2
9             9        3        3
```

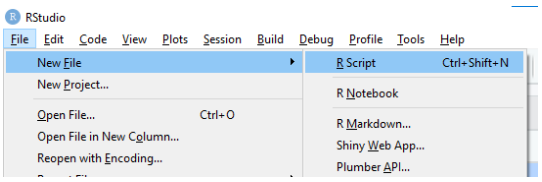
Kungla rahva laulus täishäälikute arvud sulghäälikute arvu kaupa

```
sonad %>% filter(lugu=="kungla") %>% group_by(sulghaalikuid) %>%
  summarise(taish_keskm=mean(taishaalikuid), taish_mediaan=median(taishaalikuid))
```

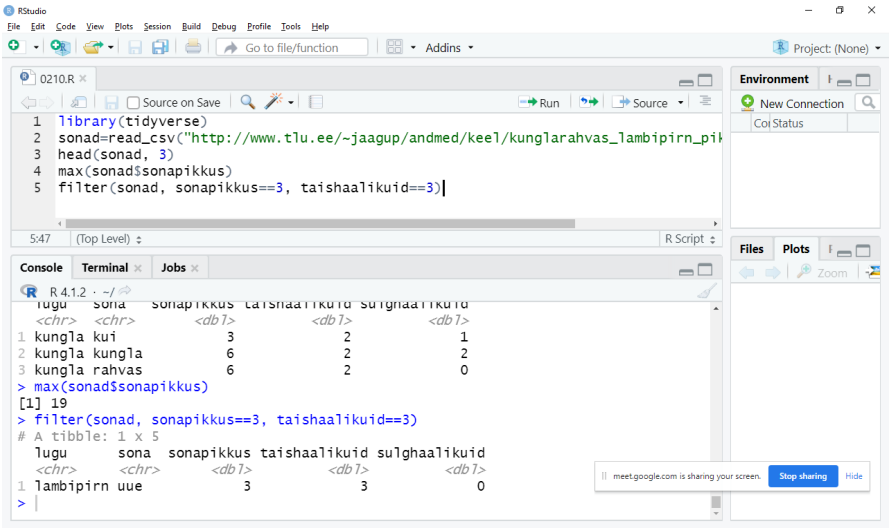
```
# A tibble: 4 x 3
  sulghaalikuid taish_keskm taish_mediaan
          <int>       <dbl>         <dbl>
1             0        2.19             2
2             1        2.45             2
3             2        1.88             2
4             3        2.5              2.5
```

Käsklused failist

Uuesti samade käskude käivitamiseks on kasulik nad eraldi faili koondada ja salvestada. Seda saab teha R Script-i nime all.



Faili tasub panna käsud nõnda, et ka vajalikud paketid ja andmed alguses sisse loetaks. Siis saab kogu jada sõltumatult käivitada.



Kood kopeerituna

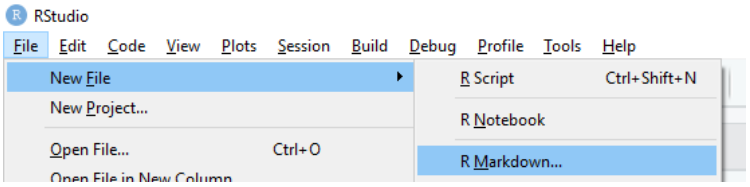
```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
head(sonad, 3)
max(sonad$sonapikkus)
filter(sonad, sonapikkus==3, taishaalikuid==3)
```

Käivitades pannakse tööle siis käsud järgemööda ning on näha ka nende töö tulemus.

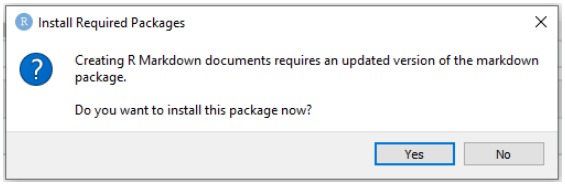
```
> library(tidyverse)
>
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
Rows: 672 Columns: 5
 0s-- Column specification ----------------------------------------------------------
Delimiter: ","
chr (2): lugu, sona
dbl (3): sonapikkus, taishaalikuid, sulghaalikuid

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> head(sonad, 3)
# A tibble: 3 x 5
  lugu    sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>   <chr>        <dbl>         <dbl>         <dbl>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
> max(sonad$sonapikkus)
[1] 19
> filter(sonad, sonapikkus==3, taishaalikuid==3)
# A tibble: 1 x 5
  lugu       sona  sonapikkus taishaalikuid sulghaalikuid
  <chr>      <chr>      <dbl>         <dbl>         <dbl>
1 lambipirn uue              3             3             0
```

RMarkdown

Veidi mitmekülgsema kujundusvormingu pakub R Markdown



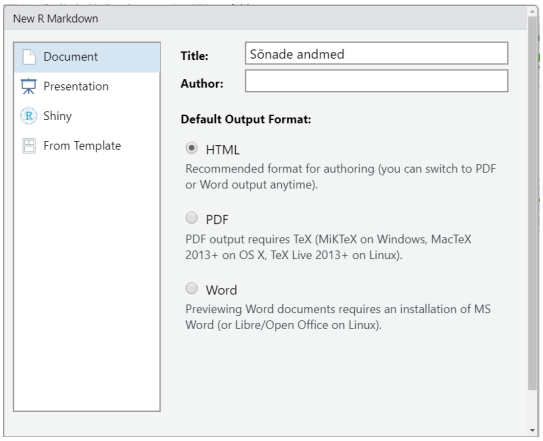
Esimesel käivitusel küsitakse, et kas vastav pakett on vaja installida.



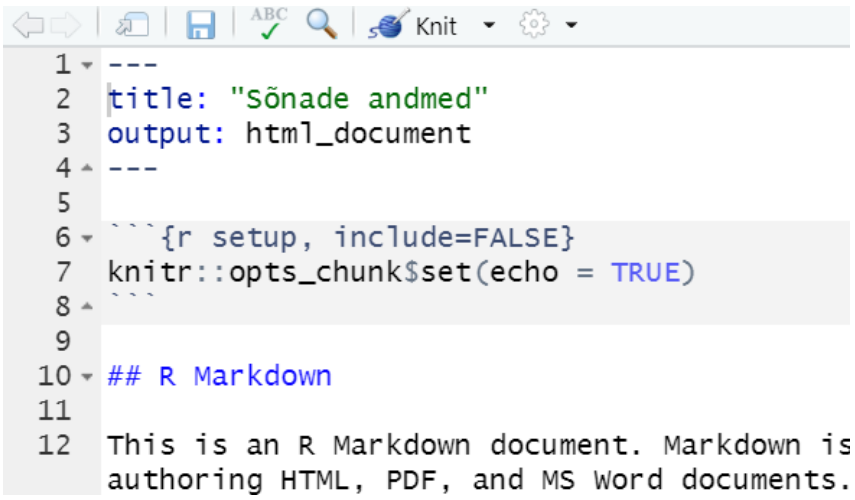
Mõnikord ei taha graafilisest keskkonnast installimine õnnestuda, kuid toimib käsurealt antud sama sisuga käsklus

```
install.packages("rmarkdown")
```

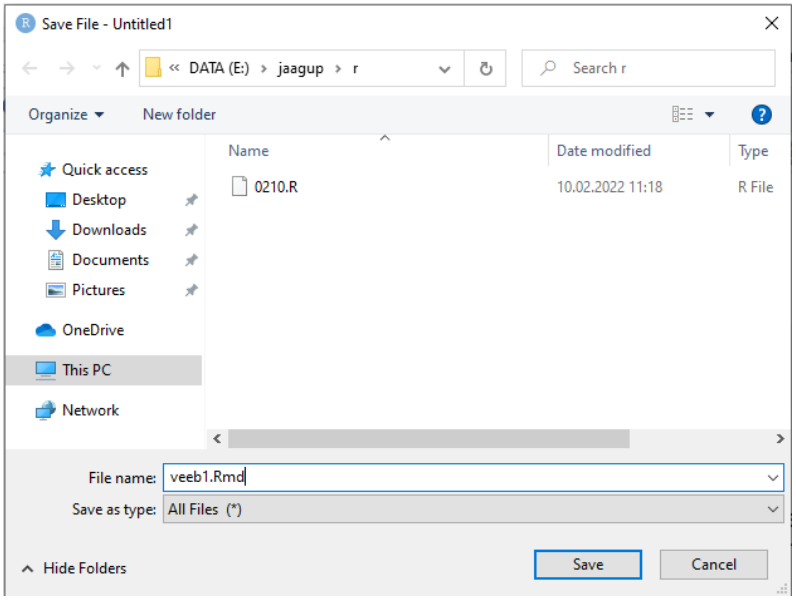
Valimisel tuleb ette aken lehe pealkirja sisestamiseks. Tüübivorminguks soovitav jätta HTML. PDF nõuab gigabaidise Tex-paketi installi, ka Wordi kasutamine on kapriisseks osutunud.



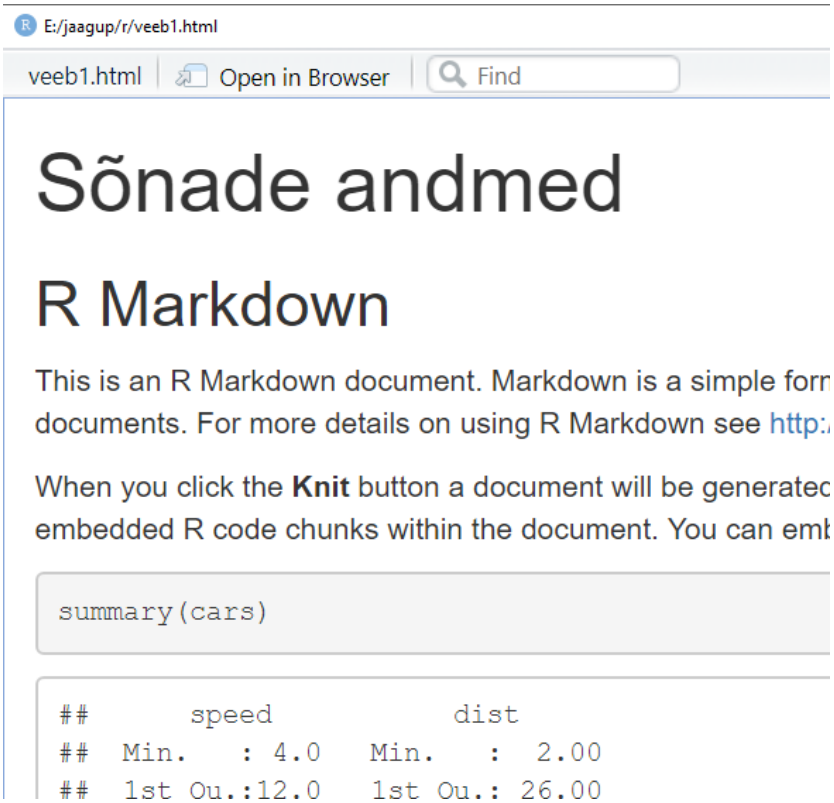
Tulemusena tekib eeltäidetud fail näitandmetega.



See tasub salvestada, laiendiks .Rmd



Knit-nupule vajutades genereeritakse faili põhjal HTML ning näidatakse seda



Edasi vaja omale sobilikud andmed sisse kirjutada. Kolme ülakoma ja r-iga algavate plokkide sisse saab panna programmikoodi, plokkidest väljas olev tekst näidatakse lihtsalt väljundisse. Kahe trelliga saab välja tuua pealkirjad.

````
---
title: "Sõnade andmed"
output: html_document
---


## Katsetused sõnade andmetega

Kõigepealt tuleb pakett ja andmed sisse lugeda

```{r, message=FALSE}
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Esimesed kolm rida tabelist

```{r}
head(sonad, 3)
```

Suurim sõnapikkus
```{r}
max(sonad$sonapikkus)
```

Ainult täishäälikutest koosnevad kolmetähelised sõnad
````

```{r}
filter(sonad, sonapikkus==3, taishaalikuid==3)
```

Uuesti `knit`-käsklusega käivitades saab juba oma andmetega lehe.

Sõnade andmed

Katsetused sõnade andmetega

Kõigepealt tuleb pakett ja andmed sisse lugeda

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Esimesed kolm rida tabelist

```
head(sonad, 3)
```

```
## # A tibble: 3 x 5
##   lugu   sona   sonapikkus taishaalikuid sulghaalikuid
##   <chr>  <chr>       <dbl>         <dbl>         <dbl>
## 1 kungla kui             3             2             1
## 2 kungla kungla          6             2             2
## 3 kungla rahvas          6             2             0
```

Suurim sõnapikkus

```
max(sonad$sonapikkus)
```

```
## [1] 19
```

Ainult täishäälikutest koosnevad kolmetähelised sõnad

```
filter(sonad, sonapikkus==3, taishaalikuid==3)
```

```
## # A tibble: 1 x 5
##   lugu      sona  sonapikkus taishaalikuid sulghaalikuid
##   <chr>     <chr>      <dbl>         <dbl>         <dbl>
## 1 lambipirn uue            3             3             0
```

Harjutus

- Tehke näited läbi
- Arvutage aritmeetiline keskmine ja mediaan täishäälikute kohta kummagi laulu sõnades - salvestage vajalikud käsud eraldi R-faili (koos library ja read_csv-ga). Salvestage fail. Sulgege R. Avage uuesti. Lugege fail ja käivitage
- Vormistage sama ülesanne RMarkdowni abil

select

library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")

```
sonad

# A tibble: 672 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <dbl>        <dbl>         <dbl>
1 kungla kui              3            2             1
2 kungla kungla           6            2             2
3 kungla rahvas           6            2             0



select(sonad, sona, sonapikkus)

# A tibble: 672 x 2
  sona    sonapikkus
  <chr>        <dbl>
1 kui              3
2 kungla           6
3 rahvas           6

> sonad %>% select(sona, sonapikkus)
# A tibble: 672 x 2
  sona    sonapikkus
  <chr>        <dbl>
1 kui              3
2 kungla           6
3 rahvas           6


> sonad %>% select(-lugu)
# A tibble: 672 x 4
  sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>        <dbl>        <dbl>         <dbl>
1 kui              3            2             1
2 kungla           6            2             2
3 rahvas           6            2             0


> sonad %>% select(-c(lugu, sulghaalikuid))
# A tibble: 672 x 3
  sona    sonapikkus taishaalikuid
  <chr>        <dbl>        <dbl>
1 kui              3            2
2 kungla           6            2
3 rahvas           6            2
```

```
> sonad %>% select(sonapikkus, everything())
# A tibble: 672 x 5
  sonapikkus lugu   sona    taishaalikuid sulghaalikuid
       <dbl> <chr>  <chr>          <dbl>         <dbl>
1          3 kungla kui                2             1
2          6 kungla kungla             2             2
3          6 kungla rahvas             2             0


> sonad %>% select(sona:taishaalikuid)
# A tibble: 672 x 3
  sona    sonapikkus taishaalikuid
  <chr>        <dbl>         <dbl>
1 kui              3             2
2 kungla           6             2
3 rahvas           6             2

> sonad %>% rename(loonimi=lugu)
# A tibble: 672 x 5
  loonimi sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>   <chr>        <dbl>         <dbl>         <dbl>
1 kungla  kui              3             2             1
2 kungla  kungla           6             2             2
3 kungla  rahvas           6             2             0

> sonad %>% mutate(muud_kaash=sonapikkus-taishaalikuid-sulghaalikuid)
# A tibble: 672 x 6
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid muud_kaash
   <chr>  <chr>        <int>         <int>         <int>      <int>
 1 kungla kui              3             2             1          0
 2 kungla kungla           6             2             2          2
 3 kungla rahvas           6             2             0          4


> sonad %>% transmute(sona=sona, muud_kaash=sonapikkus-taishaalikuid-sulghaalikuid)
# A tibble: 672 x 2
  sona    muud_kaash
  <chr>        <dbl>
1 kui              0
2 kungla           2
3 rahvas           4
```

summarise_if

Kõikidest arvulistest tulpadest ühe käsu abil võetud keskmine

```
sonad %>% group_by(lugu) %>% summarise_if(is.numeric, mean)

# A tibble: 2 x 4
  lugu      sonapikkus taishaalikuid sulghaalikuid
  <chr>          <dbl>         <dbl>         <dbl>
1 kungla          4.76          2.27          0.68
2 lambipirn       5.89          2.67          1.32
```

Aritmeetiline keskmine ja mediaan kõikidest arvulistest tulpadest

```
> sonad %>% group_by(lugu) %>% summarise_if(is.numeric, c(aritmkesk=mean, mediaan=median))
# A tibble: 2 x 7
  lugu  sonapikkus_arit~ taishaalikuid_a~ sulghaalikuid_a~ sonapikkus_medi~
taishaalikuid_m~
  <chr>            <dbl>            <dbl>            <dbl>            <int>
<int>
1 kung~             4.76             2.27             0.68                4
2
2 lamb~             5.89             2.67             1.32                5
2
# ... with 1 more variable: sulghaalikuid_mediaan <int>
```

Kuna tulpade nimed läksid pikaks ja tulemus ei mahtunud ekraanile, siis tekstiaknas näidati seda lühemalt. Graafilise lahenduse puhul võimalik vaadata sama tulemust mugavamalt View-käsu abil

```
> View(sonad %>% group_by(lugu) %>% summarise_if(is.numeric, c(aritmkesk=mean,
mediaan=median)))
```

	lugu	sonapikkus_aritmkesk	taishaalikuid_aritmkesk	sulghaalikuid_aritmkesk	sonapikkus_mediaan	taishaalikuid_m
1	kungla	4.760000	2.266667	0.680000	4	2
2	lambipirn	5.889447	2.666667	1.319933	5	2

Üheks mooduseks tulbad paremini silmade ette meelitada on nad ümber nimetada. Siis tuleb alt välja, et sp_a on sõnapikkuse aritmeetiline keskmine ning th_med on täishäälikute arvu mediaan

```
sonad %>% rename(sp=sonapikkus, th=taishaalikuid, sh=sulghaalikuid) %>%
    summarise_if(is.numeric, c(a=mean, med=median))

# A tibble: 1 x 6
   sp_a  th_a  sh_a sp_med th_med sh_med
  <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>
1  5.76  2.62  1.25      5      2      1
```

Kui sama tulemust tahta kummagi teksti kohta eraldi, siis tuleb andmestik kõigepealt loo järgi grupeerida

```
sonad %>%
  group_by(lugu) %>%
  rename(sp=sonapikkus, th=taishaalikuid, sh=sulghaalikuid) %>%
  summarise_if(is.numeric, c(a=mean, med=median))
```

```
# A tibble: 2 x 7
  lugu       sp_a  th_a  sh_a sp_med th_med sh_med
  <chr>     <dbl> <dbl> <dbl>  <int>  <int>  <int>
1 kungla     4.76  2.27  0.68      4      2      1
2 lambipirn  5.89  2.67  1.32      5      2      1
```

Loo nimi jääb rühma eristavaks tunnuseks. Kui select-käsuga eemaldan sõnade tulba, siis kõik arvutamisel alles jäävad tulbad ongi juba arvulised ning kasutada võin summarise_if-i asemel käsku summarise_all

```
sonad %>% select(-sona) %>% group_by(lugu) %>% summarise_all(mean)
```

```
# A tibble: 2 x 4
  lugu      sonapikkus taishaalikuid sulghaalikuid
  <chr>          <dbl>         <dbl>         <dbl>
1 kungla          4.76          2.27          0.68
2 lambipirn       5.89          2.67          1.32
```

Kui ei rühmita ja jätab alles kõik arvulised tulbad, siis saab sama käsklust ka neile mugavasti rakendada

```
sonad %>% select(-c(lugu, sona)) %>% summarise_all(mean)
```

```
# A tibble: 1 x 3
  sonapikkus taishaalikuid sulghaalikuid
       <dbl>         <dbl>         <dbl>
1       5.76          2.62          1.25
```

Sama tulemus ka juhul, kui välja küsida arvulised tulbad ning siis neile funktsioon rakendada

```
> sonad %>% select_if(is.numeric) %>% summarise_all(mean)
# A tibble: 1 x 3
  sonapikkus taishaalikuid sulghaalikuid
       <dbl>         <dbl>         <dbl>
1       5.76          2.62          1.25
```

Võib ka mitu arvutust korraga ette anda

```
> sonad %>% select_if(is.numeric) %>% summarise_all(c(a=mean, med=median))
# A tibble: 1 x 6
  sonapikkus_a taishaalikuid_a sulghaalikuid_a sonapikkus_med taishaalikuid_med sulghaalikuid_med
         <dbl>           <dbl>           <dbl>          <dbl>             <dbl>              <dbl>
1         5.76            2.62            1.25              5                 2                  1
```

Tulpade ümbernimetamine

Kõigepealt näide tekstist lõigu välja võtmiseks. Käsklus str_sub eraldab lõigu soovitud kohtade vahel, praegu "tere" juures esimesest teise täheni.

```
> str_sub("tere", 1, 2)
[1] "te"
```

Funktsiooni saab kõigi tulpade ümbernimetamiseks rakendada käsu rename_all abil. Punkt täistab parasjagu aktiivset tulbanime. Sellest jäetakse alles vaid kaks esimest tähte ning ongi nimetused lühemad: lu=lugu, so on nii sõna kui sõnapikkus.

```
> sonad %>% rename_all(funs(str_sub(., 1, 2)))
# A tibble: 672 x 5
   lu     so         so    ta    su
   <chr>  <chr>   <int> <int> <int>
 1 kungla kui         3     2     1
 2 kungla kungla      6     2     2
```

Et aga `is.numeric`-funktsiooni abil vaid arvulised tulbad alles jäävad, siis saab tehte ikka arusaadavalt ette võtta

```
sonad %>%
  select_if(is.numeric) %>%
  rename_all(funs(str_sub(., 1, 2))) %>%
  summarise_all(c(a=mean, med=median))


# A tibble: 1 x 6
   so_a  ta_a  su_a so_med ta_med su_med
  <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>
1  5.76  2.62  1.25      5      2      1
```

Harjutus

- Kuva laulude kaupa kõigi arvuliste tulpade vähim ja suurim väärtus
- Kuva sulghäälikute arvu kaupa sõnas viietäheliste ja lühemate sõnade täishäälikute vähim ja suurim arv

Suhtelised arvutused

Märgatavalt erinevaid algandmeid saab võrrelda, kui võrrelda suhteid väärtuste vahel.

Alustuseks juurde tulp muudest kaashäälikutest, mis pole sulghäälikud

```
> sonad %>% mutate(muud_kaash=sonapikkus-taishaalikuid-sulghaalikuid)
# A tibble: 672 x 6
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid muud_kaash
   <chr>  <chr>        <int>         <int>         <int>      <int>
 1 kungla kui              3             2             1          0
 2 kungla kungla           6             2             2          2
 3 kungla rahvas           6             2             0          4
```

Sõnapikkuste keskmise arvutus

```
 > mean(sonad$sonapikkus)
[1] 5.763393
```

Nüüd saab juba sõnapikkust võrrelda tabeli keskmise sõnapikkusega. Paistab, et kolmetäheline "kui" on veidi üle poole keskmise pikkuse ning kuuetäheline "kui" veidi üle sõnade keskmise pikkuse

```
> sonad %>% mutate(suhtelinesonapikkus=sonapikkus/mean(sonapikkus))
# A tibble: 672 x 6
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid suhtelinesonapikkus
   <chr>  <chr>        <int>         <int>         <int>               <dbl>
 1 kungla kui              3             2             1               0.521
 2 kungla kungla           6             2             2               1.04
 3 kungla rahvas           6             2             0               1.04
 4 kungla kuldsel          7             2             2               1.21
 5 kungla aal              3             2             0               0.521
```

Lugude kaupa võrreldes paistab, et Kungla rahva kõik sõnad on keskmiselt veidi rohkem kui ühe tähe võrra lühemad

```
> sonad %>% group_by(lugu) %>% summarise(keskminepikkus=mean(sonapikkus))
# A tibble: 2 x 2
  lugu      keskminepikkus
  <chr>              <dbl>
1 kungla              4.76
2 lambipirn           5.89
```

Nii põhjust arvutada sõnade suhteline pikkus võrrelduna vastava teksti sõnade keskmise pikkusega

```
> sonad %>% group_by(lugu) %>% mutate(suhtelinekeskminepikkus=sonapikkus/mean(sonapikkus))
# A tibble: 672 x 6
# Groups:   lugu [2]
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid suhtelinekeskminepikkus
   <chr>  <chr>        <int>         <int>         <int>                   <dbl>
 1 kungla kui              3             2             1                   0.630
 2 kungla kungla           6             2             2                   1.26
 3 kungla rahvas           6             2             0                   1.26
 4 kungla kuldsel          7             2             2                   1.47
 5 kungla aal              3             2             0                   0.630
```

Väljundit vaadates paistab, et andmed on jäänud grupeerituks lugude kaupa. Et see võib hilisemate arvutuste juures üllatusi põhjustada, siis lisatakse järgmisele käsule ungroup ja

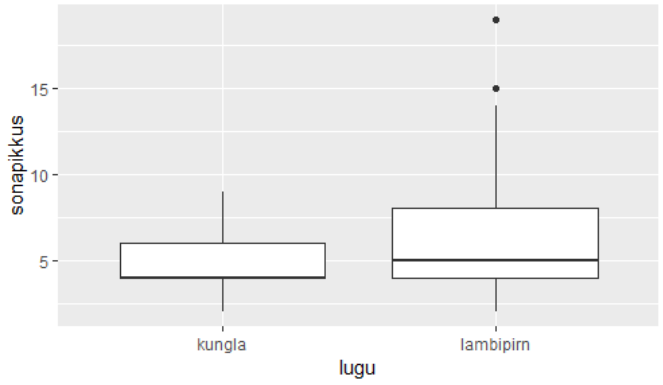
jäetakse ühtlasi alles vaid edaspidi vajalikud tulbad. Tähelepanu juhtimiseks, et summarise arvutab tulemus rühmade kaupa, nii et iga rühma (näiteks teksti) kohta jääb üks rida. Käsk mutate aga jätab kõik read alles ning lihtsalt arvutuse sees saab rühma pealt kokku arvutatud tulemusi kasutada.

```
sonad %>% group_by(lugu) %>%
  mutate(suhtelinekeskminepikkus=sonapikkus/mean(sonapikkus),
         keskminepikkus=mean(sonapikkus)) %>%
  ungroup() %>%
  select(lugu, sona, sonapikkus, keskminepikkus, suhtelinekeskminepikkus)
```

```
# A tibble: 672 x 5
   lugu   sona    sonapikkus keskminepikkus suhtelinekeskminepikkus
   <chr>  <chr>        <int>          <dbl>                   <dbl>
 1 kungla kui              3           4.76                   0.630
 2 kungla kungla           6           4.76                   1.26
 3 kungla rahvas           6           4.76                   1.26
 4 kungla kuldsel          7           4.76                   1.47
 5 kungla aal              3           4.76                   0.630
```

Andmete illustreerimiseks väike karpdiagramm: x-teljel loo nimi ning y-teljel sõnapikkuste jaotus selles loos.

```
ggplot(sonad, aes(lugu, sonapikkus)) + geom_boxplot()
```



Harjutus

- Leidke kummagi teksti suurim sõnapikkus
- Väljastage iga sõna kohta sõna pikkuse suhe selle teksti pikima sõna pikkusega
- Väljastage kummagi teksti kohta, milline osa sõnadest on lühemad kui pool pikima sõna pikkusest

Lahendusi

Suurim sõnapikkus tekstis

```
> sonad %>% group_by(lugu) %>% summarise(suurimpikkus=max(sonapikkus))
# A tibble: 2 x 2
  lugu      suurimpikkus
  <chr>            <dbl>
1 kungla               9
2 lambipirn           19
```

Iga sõna pikkuse suhe teksti pikimasse sõnasse

```
sonad %>% group_by(lugu) %>%
  mutate(suhemaxpikkus=sonapikkus/max(sonapikkus)) %>%
  ungroup() %>%
  select(lugu, sona, sonapikkus, suhemaxpikkus)

# A tibble: 672 x 4
   lugu   sona    sonapikkus suhemaxpikkus
   <chr>  <chr>        <int>         <dbl>
 1 kungla kui              3         0.333
 2 kungla kungla           6         0.667
 3 kungla rahvas           6         0.667
 4 kungla kuldsel          7         0.778
 5 kungla aal              3         0.333
```

Sama omistatuna eraldi muutujasse

```
sonadsuhemax <-
  sonad %>% group_by(lugu) %>%
  mutate(suhemaxpikkus=sonapikkus/max(sonapikkus)) %>%
  ungroup() %>%
  select(lugu, sona, sonapikkus, suhemaxpikkus)
```

Vastuse lõpuosa teise teksti väärtuste nägemiseks

```
sonadsuhemax %>% tail()

# A tibble: 6 x 4
  lugu      sona   sonapikkus suhemaxpikkus
  <chr>     <chr>       <int>         <dbl>
1 lambipirn pea             3         0.158
2 lambipirn kuklas          6         0.316
3 lambipirn ja              2         0.105
4 lambipirn suu             3         0.158
5 lambipirn pärani          6         0.316
6 lambipirn lahti           5         0.263
```

Kungla rahvast sõnad, mille pikkus on vähem kui pool vastava teksti pikima sõna pikkusest

```
> sonadsuhemax %>% filter(lugu=="kungla" & suhemaxpikkus<0.5)
# A tibble: 38 x 4
   lugu   sona  sonapikkus suhemaxpikkus
   <chr>  <chr>      <int>         <dbl>
 1 kungla kui            3         0.333
 2 kungla aal            3         0.333
 3 kungla kord           4         0.444
 4 kungla maha           4         0.444
```

Vastavate "lühemate" sõnade arv tekstis

```
> sonadsuhemax %>% filter(lugu=="kungla" & suhemaxpikkus<0.5) %>% nrow()
[1] 38
```

Sõnade üldarv tekstis

```
> sonadsuhemax %>% filter(lugu=="kungla") %>% nrow()
[1] 75
```

Kuni poole pikkusega sõnade arvu suhe vastava teksti sõnade üldarvu

```
> 38/75
[1] 0.5066667
```

Sama teise teksti puhul

```
> sonadsuhemax %>% filter(lugu=="lambipirn" & suhemaxpikkus<0.5) %>% nrow()
[1] 528
> sonadsuhemax %>% filter(lugu=="lambipirn") %>% nrow()
[1] 597
> 528/597
[1] 0.8844221
```

Arvutus mõlema tekstiga korraga. Leiame kõigepealt tulba näitamaks, kas vastava sõna pikkus on vähem kui pool selle teksti suurimast sõnapikkusest

```
> sonadsuhemax %>% mutate(lyhem=suhemaxpikkus<0.5)
# A tibble: 672 x 5
   lugu   sona    sonapikkus suhemaxpikkus lyhem
   <chr>  <chr>        <int>         <dbl> <lgl>
 1 kungla kui              3         0.333 TRUE
 2 kungla kungla           6         0.667 FALSE
 3 kungla rahvas           6         0.667 FALSE
 4 kungla kuldsel          7         0.778 FALSE
 5 kungla aal              3         0.333 TRUE
```

Andmed grupeerituna loo ning jah/ei väärtuse kaupa ja tulemused kokku loetud

```
> sonadsuhemax %>% mutate(lyhem=suhemaxpikkus<0.5) %>%
+     group_by(lugu, lyhem) %>% summarise(kogus=n())
# A tibble: 4 x 3
# Groups:   lugu [?]
  lugu      lyhem kogus
  <chr>     <lgl> <int>
1 kungla    FALSE    37
2 kungla    TRUE     38
3 lambipirn FALSE    69
4 lambipirn TRUE    528
```

Lambipirni loo puhul paistab välja, et enamik sõnu (528) on lühemad kui pool suurimast sõnapikkusest - põhjuseks mõned hästi pikad sõnad tekstis, mis maksimumi üles viivad.

Eelmine arvutus veidi lühemalt - loetakse kokku sõnade arv, mille pikkus ületab poole suurimast sõnapikkusest tekstis

```
> sonadsuhemax %>% group_by(lugu) %>% summarise(lyhem=length(sona[suhemaxpikkus<0.5]))
# A tibble: 2 x 2
  lugu      lyhem
  <chr>     <int>
1 kungla       38
2 lambipirn   528
```

Või veel lühemalt: liidetakse kokku jah-vastused tingimusele

```
> sonadsuhemax %>% group_by(lugu) %>% summarise(lyhem=sum(suhemaxpikkus<0.5))
# A tibble: 2 x 2
  lugu      lyhem
  <chr>     <int>
1 kungla       38
2 lambipirn   528
```

Juurde ka lühemate sõnade suhtarv kogu teksti sõnade arvu suhtes

```
> sonadsuhemax %>% group_by(lugu) %>%
   summarise(lyhem=sum(suhemaxpikkus<0.5), kokku=n(), suhe=lyhem/kokku)
# A tibble: 2 x 4
  lugu      lyhem kokku  suhe
  <chr>     <int> <int> <dbl>
1 kungla       38    75 0.507
2 lambipirn   528   597 0.884
```

Joonised

Andmete esitamise juures on levinuks saanud paketi `ggplot2` joonised. Pakett kuulub juba tuttava `tidyverse` paketi koosseisu. Andmetega tegeldakse seal nutikate `tibble`-tabelite kaudu. Alustuseks näide andmeloetelust `tibble` loomiseks. Andmed võetud aadressilt http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_koik_sonaliigid.txt

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")

tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"), kogus=c(34, 14, 11,
11))
```

Väljund:

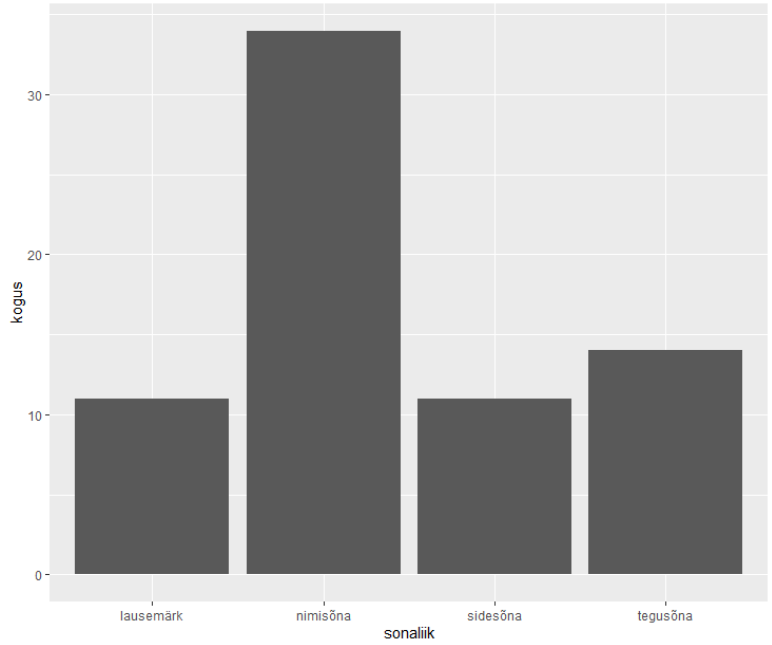
```
# A tibble: 4 x 2
  sonaliik  kogus
  <chr>     <dbl>
1 nimisõna     34
2 tegusõna     14
3 lausemärk    11
4 sidesõna     11
```

Tutvustusnäited

Andmed saab edasi suunata ggplot-ile. Käsuga `aes` (aestetics) määratakse, milliseid andmetulpi kasutatakse. Kui eraldi ei määrata, siis on esimene `x` ning teine `y` - praegusel juhul siis sonaliik ja kogus. Edasi saab plussmärgiga juurde lisada kuvatava joonise kihte. Käsklus `geom_col()` loob tulpdiagrammi.

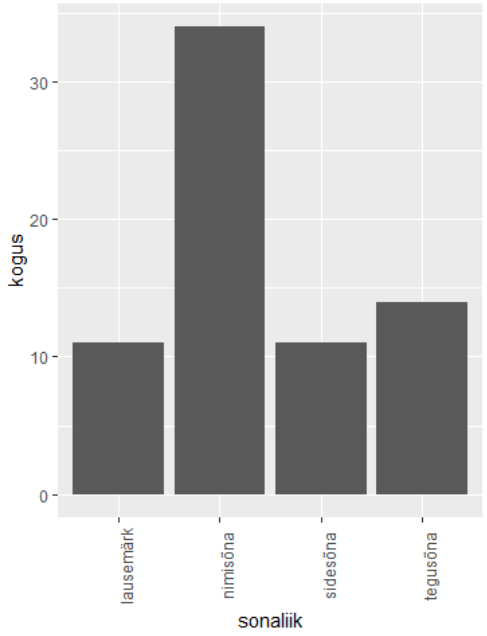
```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
    ggplot(aes(sonaliik, kogus)) + geom_col()
```

Tulemus:



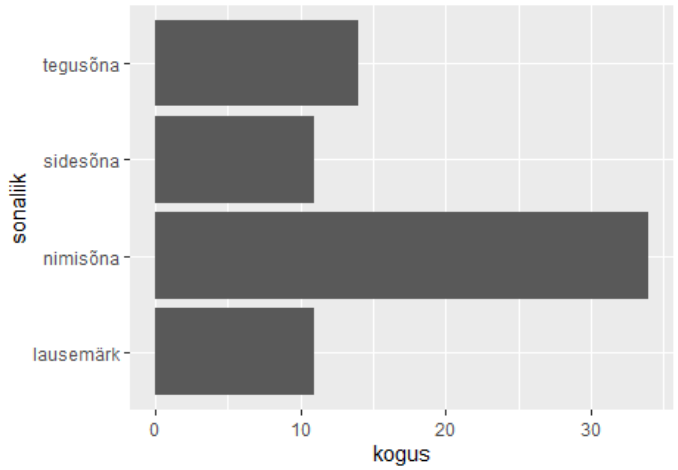
Kui tulpade seletused pikemad või neid rohkem, siis kasulik tekst panna alt üles jooksma, nii mahub rohkem. Mitmesuguste kujunduste juures aitab `theme` sättimine oma mitmesuguste parameetritega - praegusel juhul siis teatega, et x-teljel olev tekst on 90 kraadi keeratud

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(sonaliik, kogus)) + geom_col()+
    theme(axis.text.x=element_text(angle=90))
```



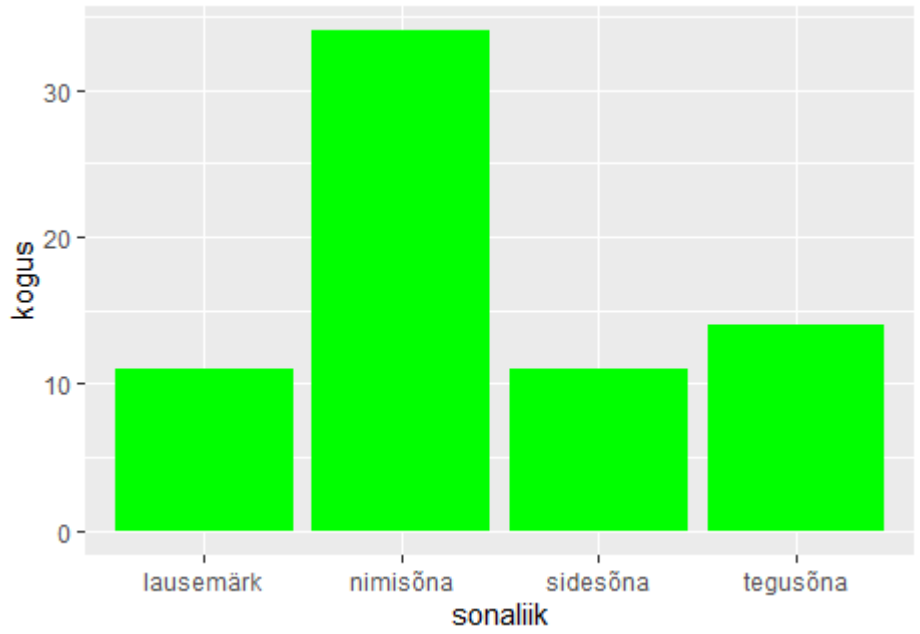
Teljed omavahel pöörata aitab `coord_flip()` - nii saab tulpi vasakult paremale vaadata

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(sonaliik, kogus)) + geom_col()+
    coord_flip()
```



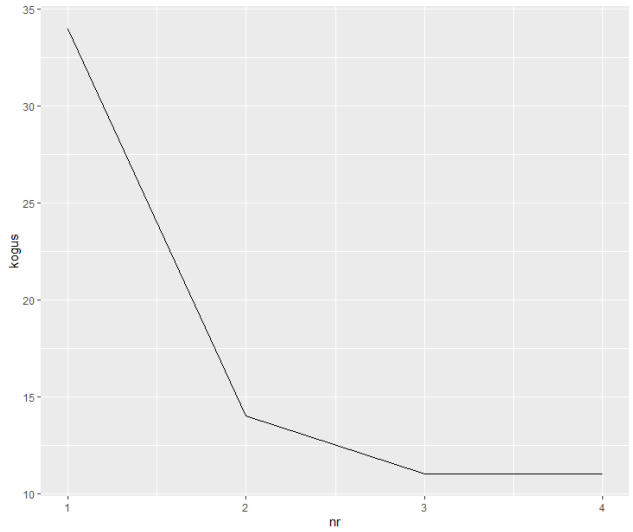
Ala sisu värvmise parameeter `fill`

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(sonaliik, kogus)) + geom_col(fill="green")
```



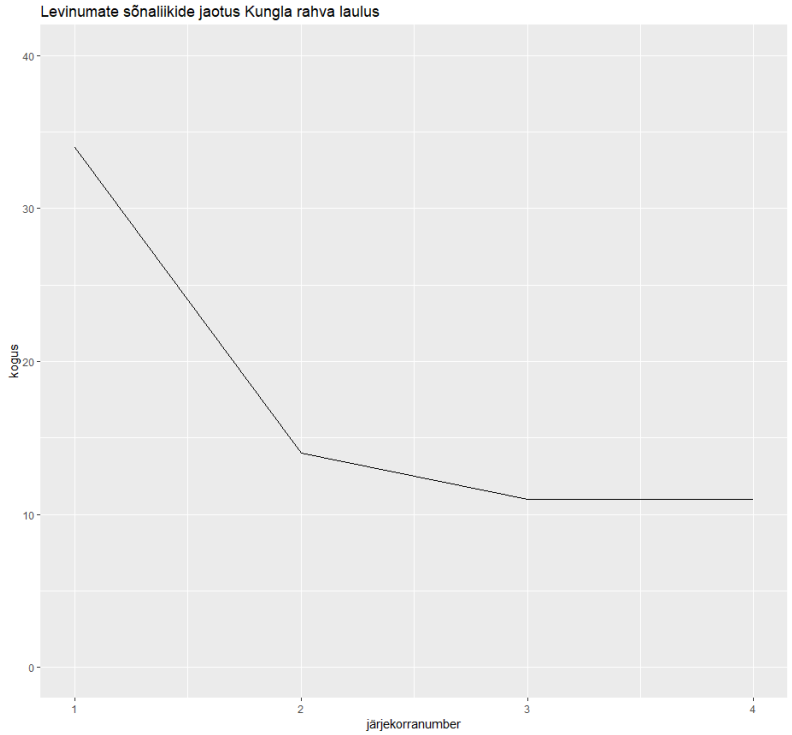
Joondiagrammi jaoks käsklus `geom_line()`. Sinna vaja väärtustele x-teljele juurde järjekorranumbrid

```
tibble(nr=c(1, 2, 3, 4), kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(x=nr, kogus)) + geom_line()
```



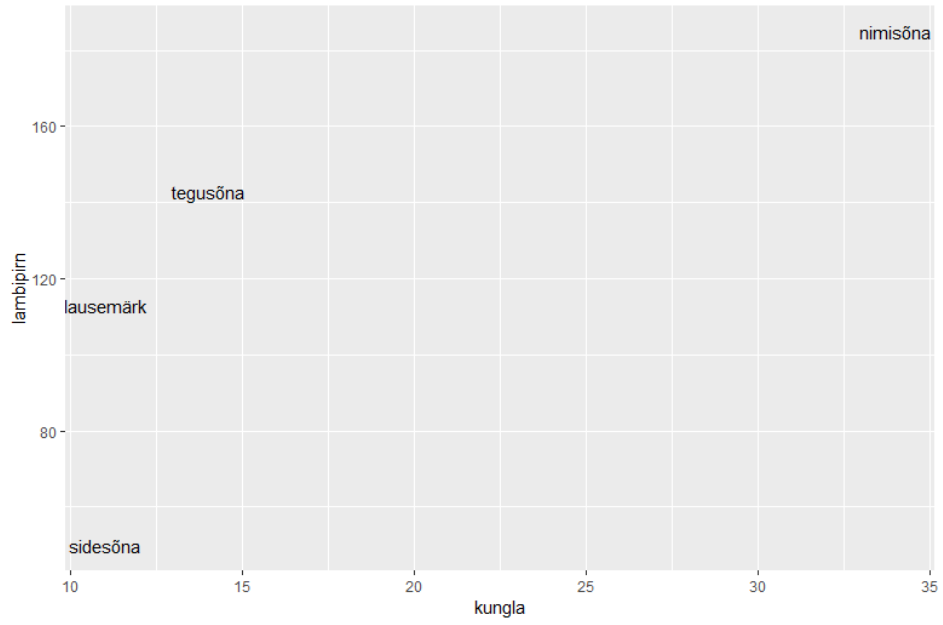
Joonisele juurde mõningased täiendused. Automaatne `y`-skaala on petlik - nagu oleks esimesi väärtusi palju kordi rohkem kui järgmisi. Käsk `ylim` määrame siin, et alampiir algab nullist, nii paistab sageduste suhe selgemalt välja. Juurde veel `xlab`, `ylab` ning `ggtitle` telgede ja pealkirja tekstide jaoks.

```
tibble(nr=c(1, 2, 3, 4),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(x=nr, kogus)) + geom_line()+
   ylim(0, 40)+  xlab("järjekorranumber")+ylab("kogus")+
   ggtitle("Levinumate sõnaliikide jaotus Kungla rahva laulus")
```



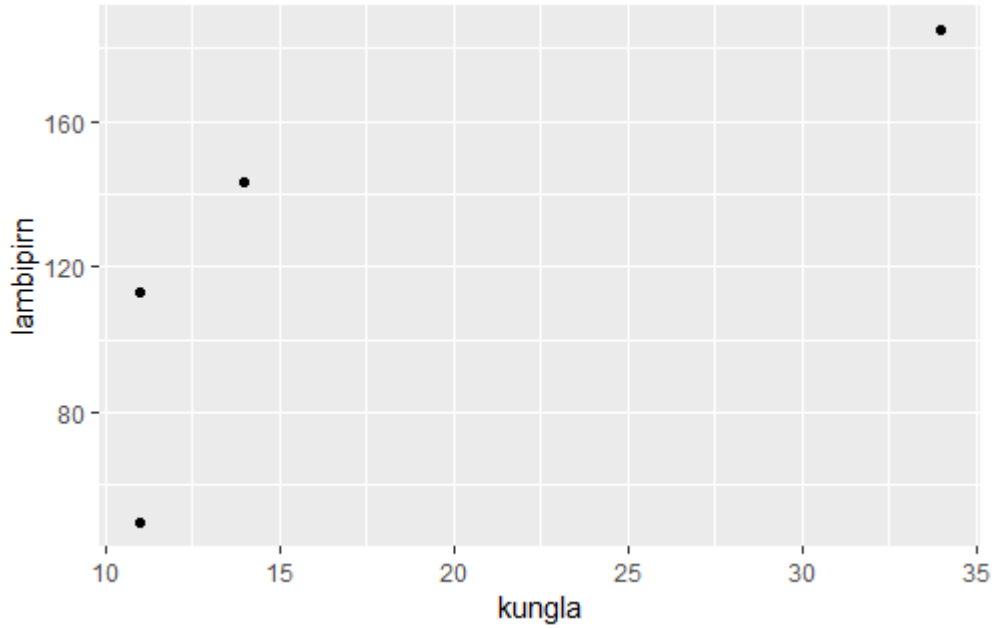
Teksti kuvamisel lisame näidatavad sõnaliigid ning lambipirni teksti võrdlusandmed. Tekst tuleb omistada aestetics-ploki tunnusena `label`

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
         kungla=c(34, 14, 11, 11),
         lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=kungla, y=lambipirn, label=sonaliik))+geom_text()
```



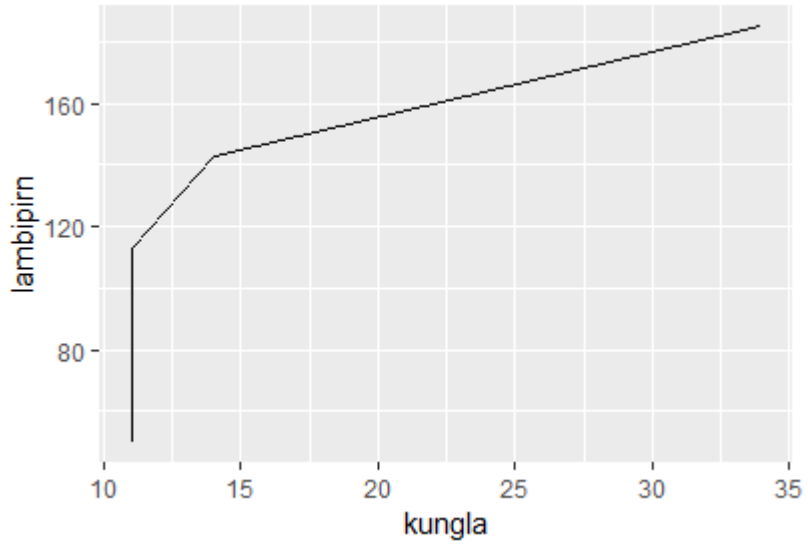
Vaid asukohapunktide märkimiseks piisab käsklus `geom_point()`

```
tibble(kungla=c(34, 14, 11, 11),
       lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=kungla, y=lambipirn))+geom_point()
```



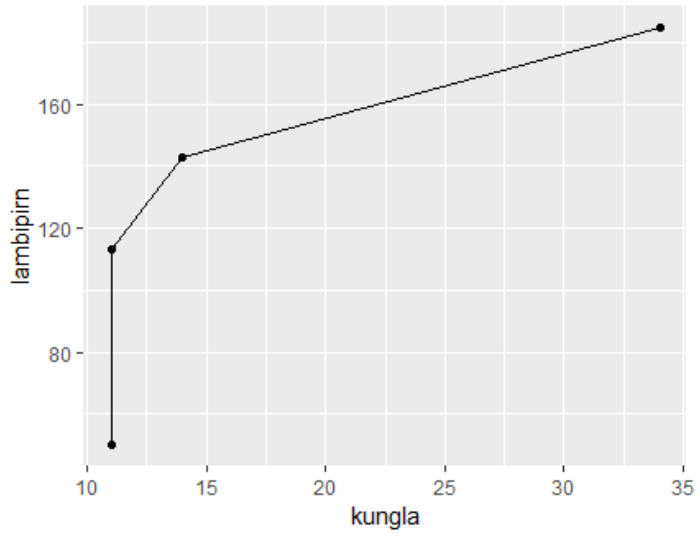
Asukohapunktide ühendamiseks sobib `geom_path()`

```
tibble(kungla=c(34, 14, 11, 11), lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=kungla, y=lambipirn))+geom_path()
```



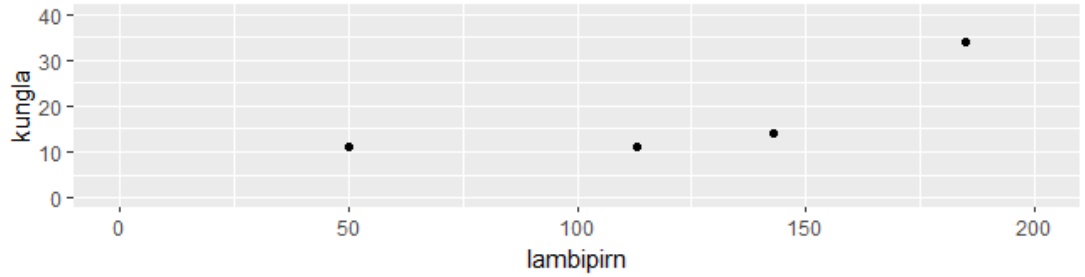
Mitme joonisekihi koos kuvamisel tuleb need lihtsalt üksteisele liita

```
tibble(kungla=c(34, 14, 11, 11), lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=kungla, y=lambipirn))+
    geom_point()+geom_path()
```



Joonise x- ja y-telje skaala ühesuguse sammu peale saamiseks aitab `coord_fixed()`

```
tibble(kungla=c(34, 14, 11, 11),
       lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=lambipirn, y=kungla))+
    geom_point()+coord_fixed()+xlim(0, 200)+ ylim(0, 40)
```

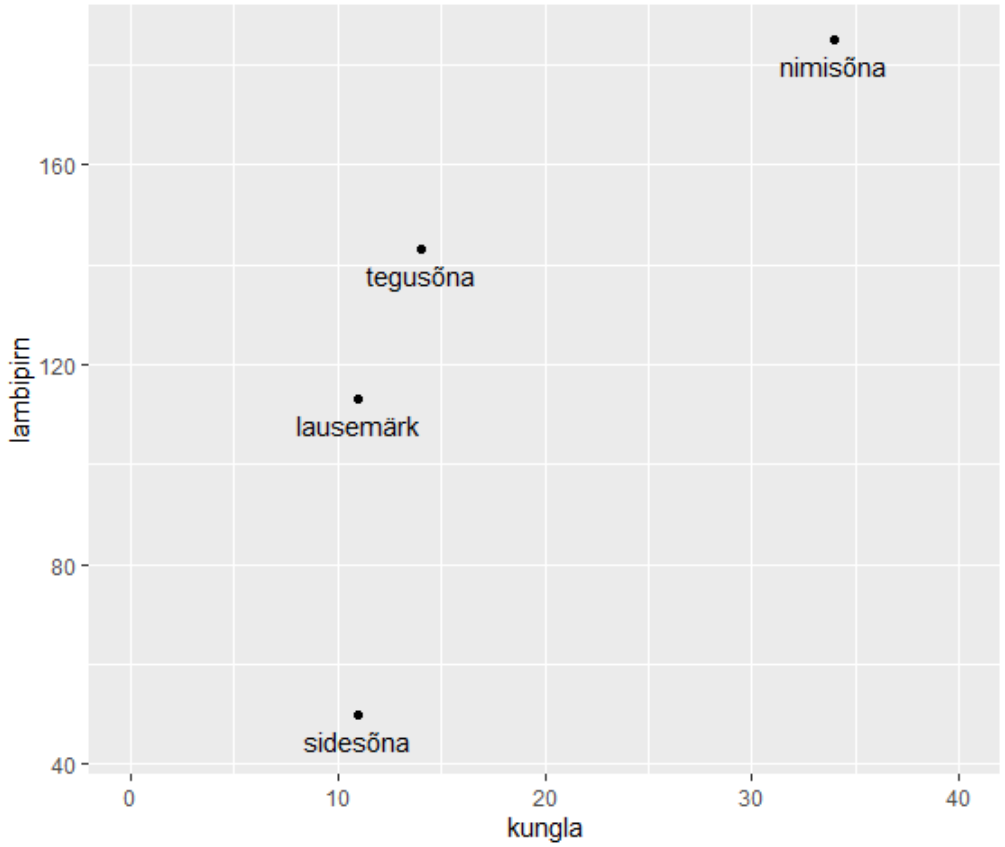


Harjutus

- Pane näited tööle
- Lisa igale poole määrsõnade arv - Kungla rahvas: 7, lambipirn: 86
- Kuva joonisel mõlema teksti sõnaliikide arv korraga tekstidena ja punktidena
- Säti skaalal, et üks esinemiskord võtab mõlema telje suunas võrdse ruumi
- Lisa telgedele seletused ja joonisele pealkiri

Tekstid nihutatud aktiivsetest punktidest viie ühiku võrra allapoole, et punktil ja tekstil mõlemil ruumi oleks.

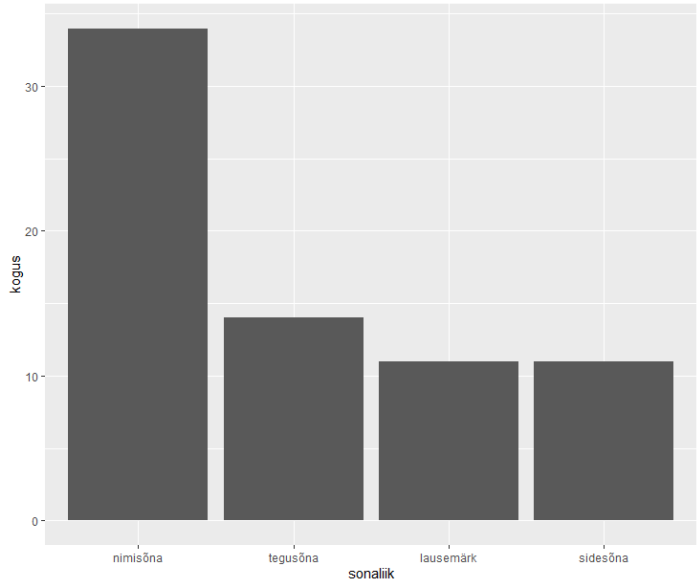
```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kungla=c(34, 14, 11, 11),
       lambipirn=c(185, 143, 113, 50)) %>%
  ggplot(aes(x=kungla, y=lambipirn, label=sonaliik))+geom_text(aes(y=lambipirn-5)) +
      geom_point() + xlim(0, 40)
```



Järjestatud tulbad

Eelnenud näitest paistis, et tulpdiagrammi tulpade selgitavad tekstid pannakse vaikimisi juhul tähestikulisse järjekorda. Erinevuste paremaks avastamiseks aga soovitatakse tulpi järjestada väärtuse järgi. Üheks järjestamise mooduseks on määrata, et tulpade nimede tunnuse tüübiks on faktor ning järjestus selline, nagu sisestatakse parameetriga `levels`

```
tibble(sonaliik=factor(c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
                levels=c("nimisõna", "tegusõna", "lausemärk", "sidesõna")),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(sonaliik, kogus)) + geom_col()
```



Eelmises näites kirjutati tulpade järjestus otse ette. Sagedamini aga läheb vaja, et tulbad näidatava tunnuse alusel järjestatakse - selle juures aitab käsklus order, mis annab väljundiks arvud näitamaks, mitmenda koha peal järjestuses vastav väärtus asub.

```
order(c(14, 34, 11, 11))
```

annab vastusteks

```
[1] 3 4 1 2
```

sest kasvavas järjekorras on üksteist esimese ja teise koha peal, 14 kolmanda ning 34 neljanda koha peal. Käsu väljund ütleb, et järjestatud rea saamiseks tuleb loetelust võtta kõigepealt kolmas, edasi neljas, siis esimene ja lõpuks teine element.

```
sonaliik=c("tegusõna", "nimisõna", "lausemärk", "sidesõna")
kogus=c(14, 34, 11, 11)
jarjestus=order(kogus)
sonaliik[jarjestus]
[1] "lausemärk" "sidesõna"  "tegusõna"  "nimisõna"
```

Kontroll, et järjestus on ikka enne vaadatud tuttav järjekord

```
> jarjestus
[1] 3 4 1 2
```

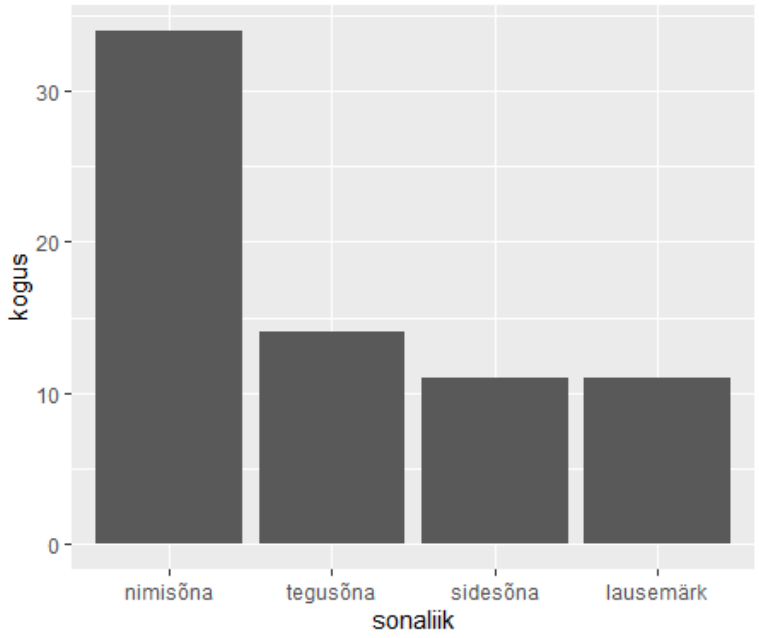
Ning silma järgi näeb, et sõnaliigid on sageduse kasvavas järjekorras lausemärk-11, sidesõna-11, tegusõna-14, nimisõna-34

Loetelu ehk praegusel juhul järjekorra ümberpööramiseks tuleb sinna ette panna `rev` - nii saab endisest 3-4-1-2 järjestusest 2-1-3-4 ning pärast nende järjekorranumbritega sõnaliike küsides on järjestus ka eelmise kasvavaga võrreldes vastupidine ehk kahanev

```
> rev(order(c(14, 34, 11, 11)))
[1] 2 1 4 3
```

Tulemus joonisel, tulba ümberarvutamiseks käsklus `mutate`:

```
tibble(sonaliik=c("tegusõna", "nimisõna", "lausemärk", "sidesõna"),
       kogus=c(14, 34, 11, 11)) %>%
        mutate(sonaliik=factor(sonaliik, levels=sonaliik[rev(order(kogus))])) %>%
      ggplot(aes(sonaliik, kogus)) + geom_col()
```



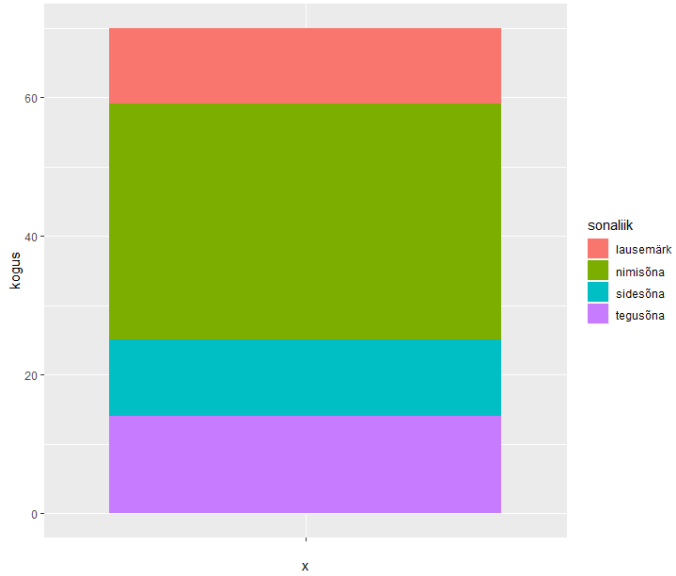
Harjutus

- Tehke näide läbi
- Lisage määrsõna (7 tk)

Sektordiagramm

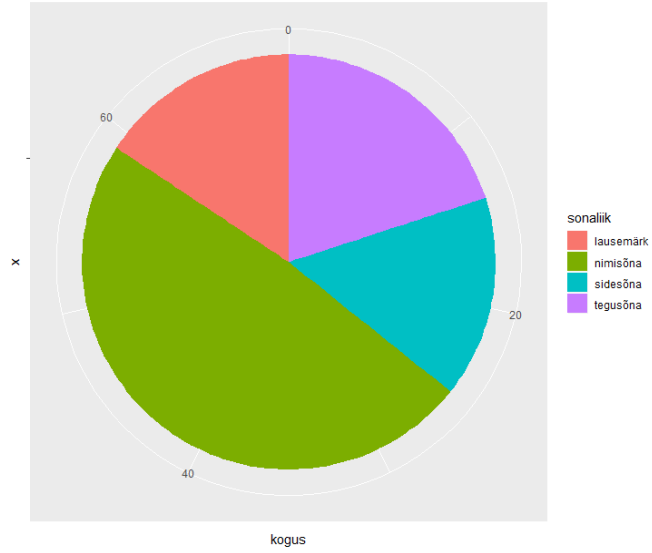
Muidu levinud sektordiagrammi tüüpi ggplot-teegis otsese käsuna ei ole. Küll aga saab selle välja meelitada tulpdiagrammi erijuhuna. Esimene samm sinnapoole on tulpdiagramm, kus kõik väärtused on üksteise otsas. X-telge eristav tunnus jäetakse tühjaks, y-suunal kuvatavad väärtused liidetakse üksteise otsa ning fill=sonaliik määrab, et kõik need kuvatakse eri värvi

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(x="", kogus, fill=sonaliik)) + geom_col()
```



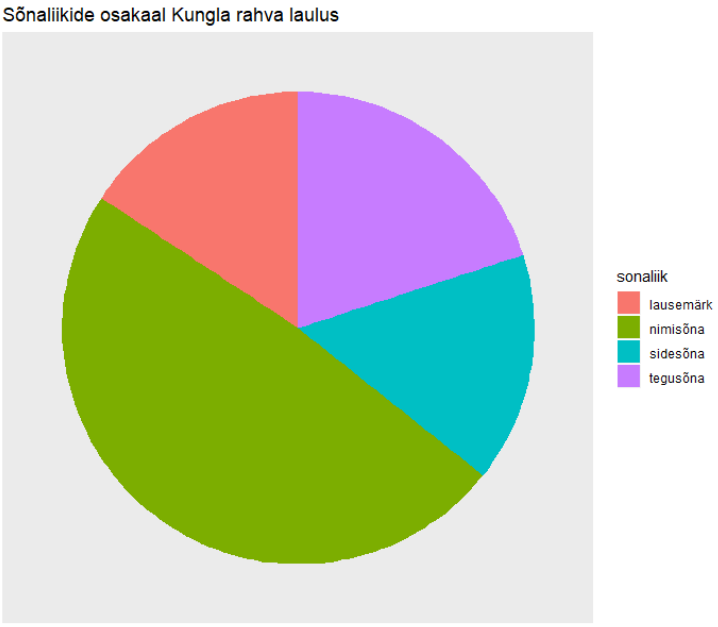
Kui juurde lisada, et joonist soovitakse polaarkoordinaatides ning painutatuna algse y-telje järgi, siis hakkab tulemus juba küllaltki sektordiagrammi moodi välja nägema

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(x="", kogus, fill=sonaliik)) + geom_col()+coord_polar("y")
```



Mõningase kohandamisega saab ka ette jäänud ja sektordiagrammi juures kasutud elemendid tühjaks muuta

```
tibble(sonaliik=c("nimisõna", "tegusõna", "lausemärk", "sidesõna"),
       kogus=c(34, 14, 11, 11)) %>%
  ggplot(aes(x="", kogus, fill=sonaliik)) + geom_col()+
    coord_polar("y")+
    theme(axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid = element_blank())+
    xlab("")+ylab("")+
    ggtitle("Sõnaliikide osakaal Kungla rahva laulus")
```



Harjutus

- Tehke näide läbi
- Lisage määrsõna (7 tk)

Andmetabeli pikk kuju

Samu andmeid saab hoida ja kuvada mitmel moel, valik tehakse sageli kasutusotstarbe järgi. Alljärgnev näide kahe siianigi võrreldud teksti andmete vahel. Laia kuju on ehk silmaga kergem haarata - näeb, väärtusi ridu ja veerge pidi kõrvuti. Pika kuju eeliseks jälle mugavam võimalus programmiga andmete poole pöörduda ka juhul, kui tunnuseid peaks hiljem juurde tulema. Tabeli tulpasid ikka kolm tükki - lugu, tunnus ja väärtus - vajalikke ridu saab lihtsalt loo või tunnuse järgi välja filtreerida.

Pikk:

	lugu	tunnus	vaartus
	<chr>	<chr>	<dbl>
1	kungla	sonapikkus	4.76
2	lambipirn	sonapikkus	5.89
3	kungla	taishaalikuid	2.27
4	lambipirn	taishaalikuid	2.67
5	kungla	sulghaalikuid	0.68
6	lambipirn	sulghaalikuid	1.32

Lai:

lugu	sonapikkus	taishaalikuid	sulghaalikuid
<chr>	<dbl>	<dbl>	<dbl>
1 kungla	4.76	2.27	0.68
2 lambipirn	5.89	2.67	1.32

Juurde ka käsud algsest sõnade tabelist sinnani jõudmiseks

```
> library(tidyverse)
```

```
> sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

```
> head(sonad)
# A tibble: 6 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
4 kungla kuldsel          7             2             2
5 kungla aal              3             2             0
6 kungla kord             4             1             2
```

Summeeritud laial kujul tabeli saame praegu, kui kõikidest arvulistest tulpadest aritmeetiline keskmine võetakse ning lugude kaupa ridadesse paigutatakse.

```
> lai_tabel <- sonad %>% group_by(lugu) %>% summarise_if(is.numeric, mean)
> lai_tabel
# A tibble: 2 x 4
  lugu      sonapikkus taishaalikuid sulghaalikuid
  <chr>          <dbl>         <dbl>         <dbl>
1 kungla          4.76          2.27          0.68
2 lambipirn       5.89          2.67          1.32
```

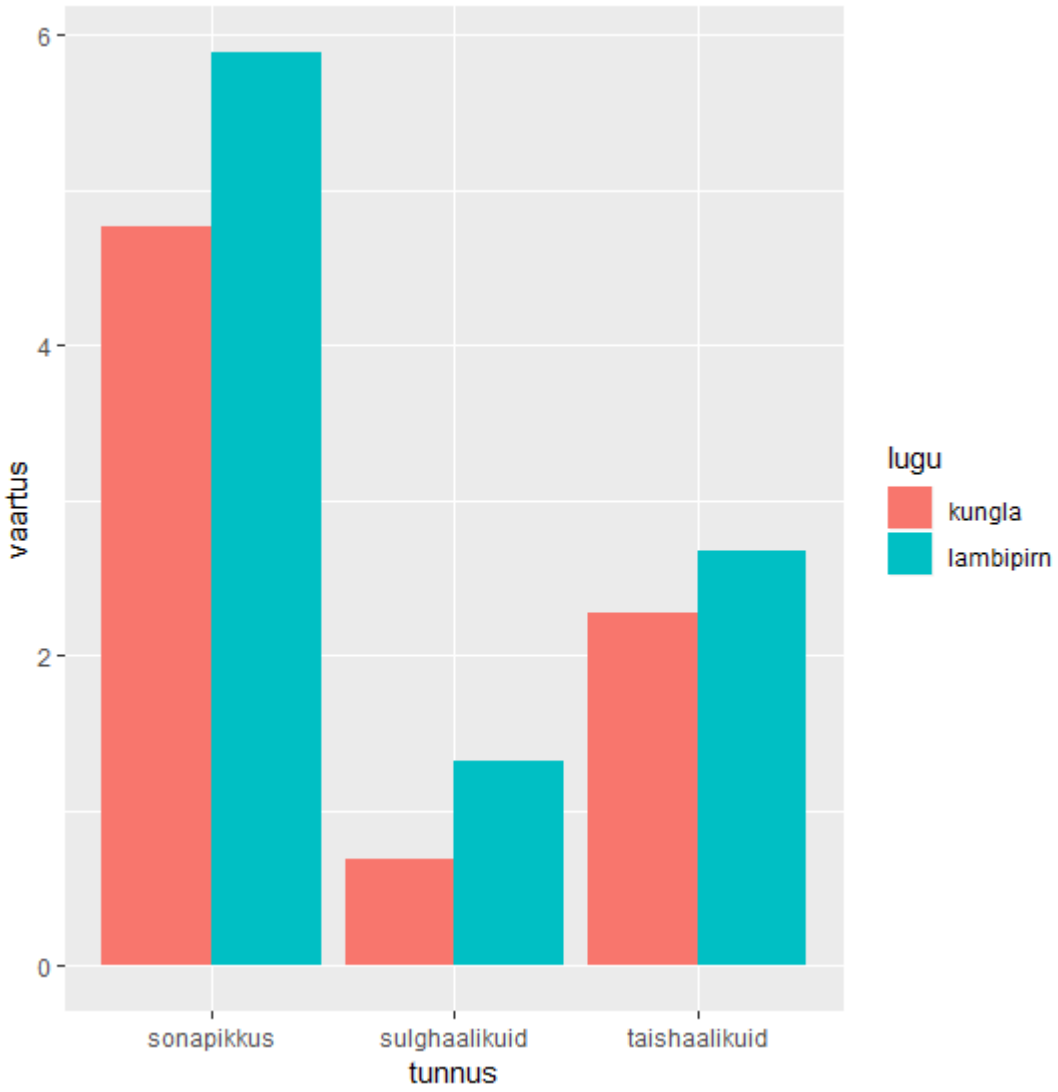
Tulpdiagramm

Jooniste koostamiseks kasutatava `ggplot`'i tulpasid loov `geom_col`-käsklus eeldab, et saab andmestiku sisse tabeli pikal kujul - sõltumata joonisel olevast tulpade arvust on sisendtabelis ikka kolm tulpa: lugu, tunnus ja väärtus. Teisendamiseks sobib käsklus gather. Esimese parameetrina on sisse antud andmestik, teine ja kolmas uute tulpade nimed ning lõpus miinusmärgiga näidatu jääb sama tunnuse tulpasid eristama.

```
> pikk_tabel <- gather(lai_tabel, tunnus, vaartus, -lugu)
> pikk_tabel
# A tibble: 6 x 3
  lugu      tunnus        vaartus
  <chr>     <chr>           <dbl>
1 kungla    sonapikkus       4.76
2 lambipirn sonapikkus       5.89
3 kungla    taishaalikuid    2.27
4 lambipirn taishaalikuid    2.67
5 kungla    sulghaalikuid    0.68
6 lambipirn sulghaalikuid    1.32
```

Pika tabeli põhjal luuakse ggplot-i tulpdiagramm. Tunnusest saab x-koordinaat ja väärtusest y-koordinaat. Täiendus position_dodge() määrab, et tulbad pannakse üksteise kõrvale.

```
 pikk_tabel %>% ggplot(aes(tunnus, vaartus, fill=lugu)) +
geom_col(position=position_dodge())
```

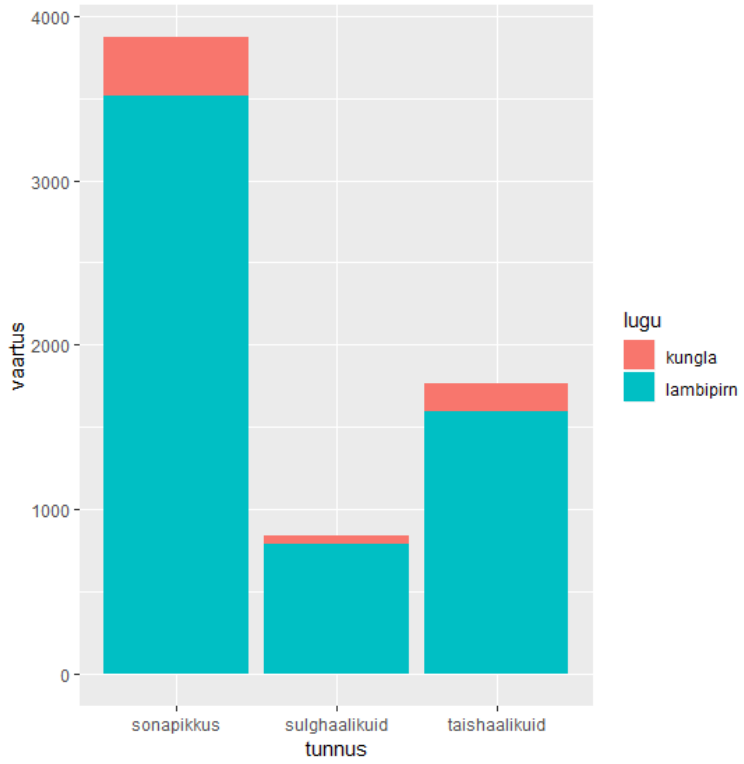


Kui näiteks ei võrreldaks keskmisi vaid üldarve, siis võib selle ära jätta ning tulba eri värvi osad pannakse üksteise peale. Siin sellise joonise koostamine etappide kaupa. Kõigepealt summa arvulistest tulpadest lugude kaupa.

```
> sonad %>% group_by(lugu) %>% summarise_if(is.numeric, sum)
# A tibble: 2 x 4
  lugu      sonapikkus taishaalikuid sulghaalikuid
  <chr>          <int>         <int>         <int>
1 kungla           357           170            51
2 lambipirn       3516          1592           788
```

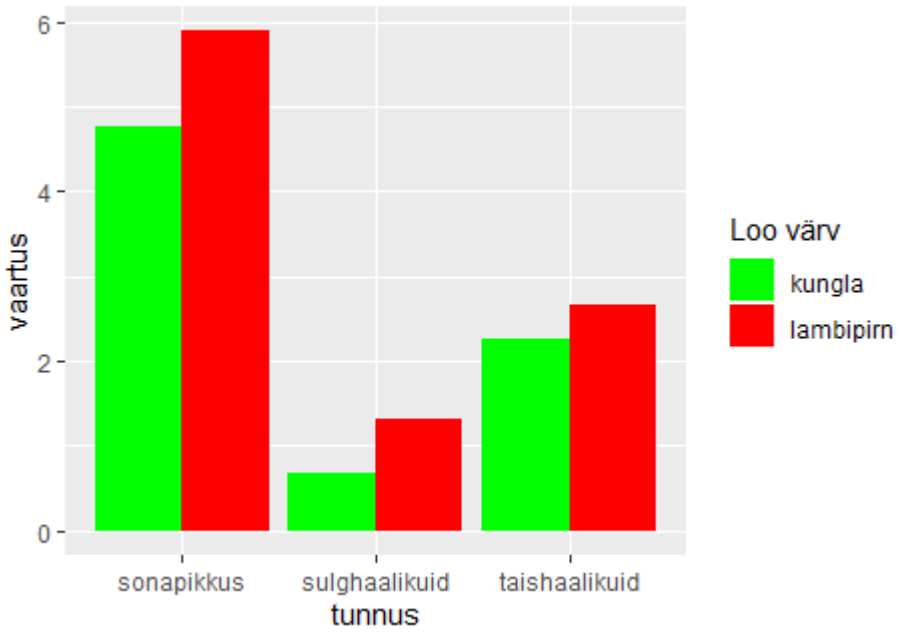
Edasi joonis, kus vastavad tulbad üksteise peal, kummagi loo andmed ise värvi.

```
sonad %>%
  group_by(lugu) %>%
  summarise_if(is.numeric, sum) %>%
  gather(tunnus, vaartus, -lugu) %>%
  ggplot(aes(tunnus, vaartus, fill=lugu))+geom_col()
```



Kui tahtmine värvid ise määrata, siis tuleb `scale_fill_manual`-i järgi ette anda, et milline väärtus millist värvi on

```
pikk_tabel %>% ggplot(aes(tunnus, vaartus, fill=lugu)) +
  geom_col(position=position_dodge()) +
   scale_fill_manual("Loo värv", values=c("kungla" ="green", "lambipirn"="red"))
```



Harjutus

- Tehke näide läbi
- Koosta sõnade andmestiku põhjal tabel, kus on kummagi loo kohta pikima sõna pikkus, sõnade keskmine pikkus ning sõnade pikkuste mediaan.
  `http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt`
- Kuva nende andmete põhjal tulpdiagramm
- Määra loole vastavate tulpade värvid ise

Lahendus

```
pikkused <- sonad %>%
  group_by(lugu) %>%
  summarise(suurim=max(sonapikkus),
            keskmine=mean(sonapikkus),
            mediaan=median(sonapikkus))
```

Andmete loetelu

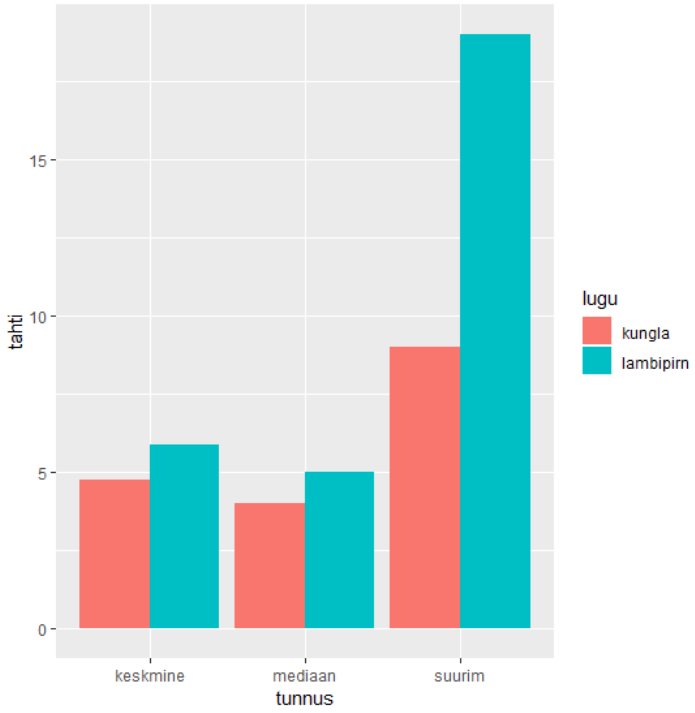
```
head(pikkused)
  lugu       suurim keskmine mediaan
  <chr>       <dbl>    <dbl>   <int>
1 kungla          9     4.76       4
2 lambipirn      19     5.89       5
```

Andmed pikale kujule

```
> pikkused %>%  gather(tunnus, tahti, -lugu)
# A tibble: 6 x 3
  lugu      tunnus   tahti
  <chr>     <chr>    <dbl>
1 kungla    suurim    9
2 lambipirn suurim   19
3 kungla    keskmine  4.76
4 lambipirn keskmine  5.89
5 kungla    mediaan   4
6 lambipirn mediaan   5
```

Kõik korraga joonisele

```
pikkused %>%
  gather(tunnus, tahti, -lugu) %>%
  ggplot(aes(tunnus, tahti, fill=lugu)) + geom_col(position=position_dodge())
```



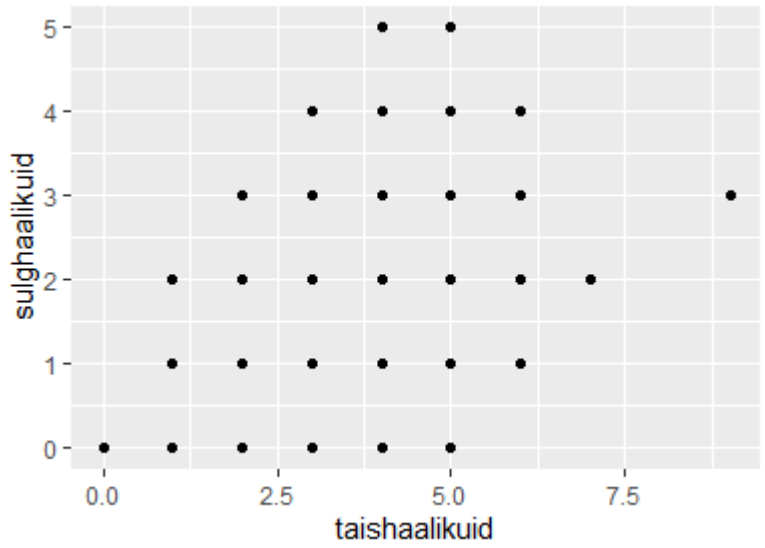
Kattuvad väärtused

Meeldetuletuseks sisse tuttavad andmed

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Täishäälikute ja sulghäälikute arvu paiknemise kohta joonis

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point()
```

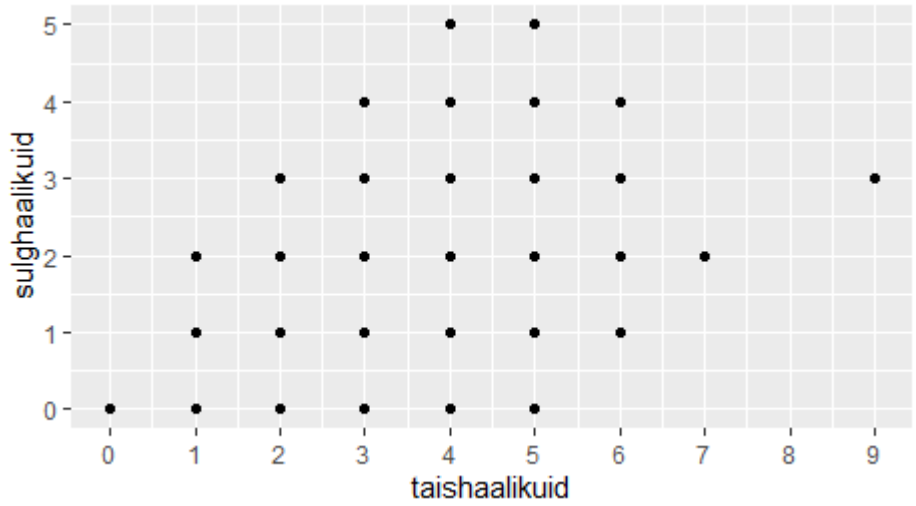


Kuna siin saab iga sümbolit olla täisarv kordi, siis mugavama loetavuse huvides ütleme ka x-telje juures, et soovitakse kirjeldust iga arvu juurde. 0:10 on lihtsalt arvude loetelu nullist kümneni

```
> 0:10
 [1]  0  1  2  3  4  5  6  7  8  9 10
```

Et suurim väärtus aga piirdub üheksa täishäälikuga sõnas ning eraldi `xlim`-käsuga joonise ulatuse pikkust ei muuda, siis kümmet ka skaalal ei paista

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point()+
  scale_x_continuous(breaks=0:10)
```

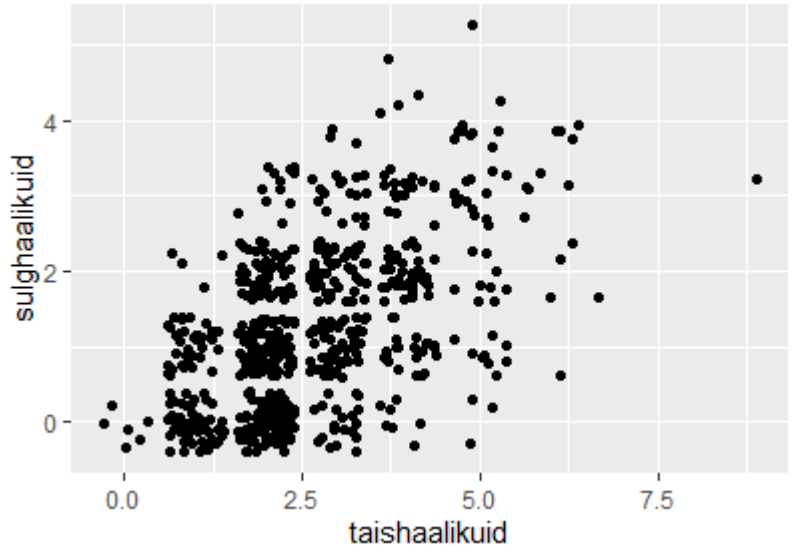


Mooduseks üksteise alla jäävaid andmeid nähtavaks teha on käsklus `geom_jitter()`. Eelmisel joonisel on nelja täishääliku ja ühe sulghääliku ristumiskohas vaid üks punkt. Arvutuste järgi on aga selliseid sõnu andmestikus tervelt 23

```
> sonad %>% filter(taishaalikuid==4, sulghaalikuid==1) %>% nrow()
[1] 23
```

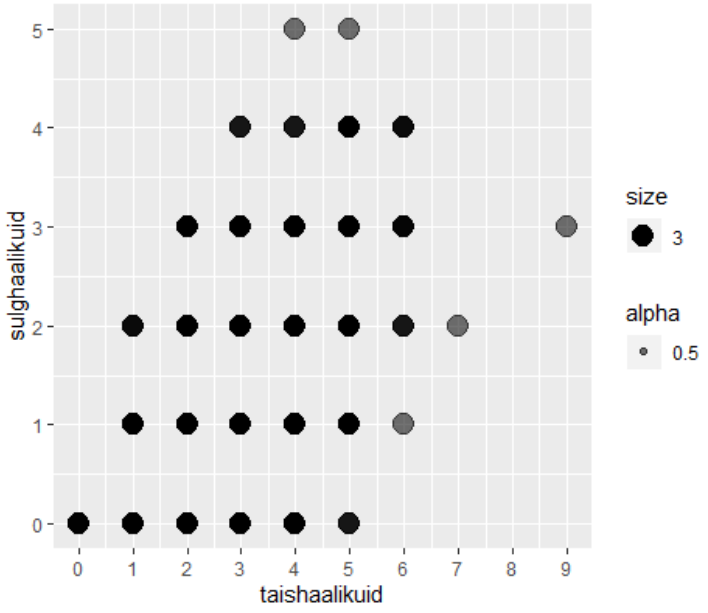
Käsklus `geom_jitter()` veidi "loksutab" neid ning nii paistavad enamik üksteise alt välja

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter()
```



Mõnevõrra aitab varju jäävate sõnade puhul läbipaistvaks muutmine - nii on üksik täpp heledam ning mitu üksteise peale sattunud täppi tumedamad

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid, alpha=0.5, size=3))+geom_point()+
  scale_x_continuous(breaks=0:10)
```



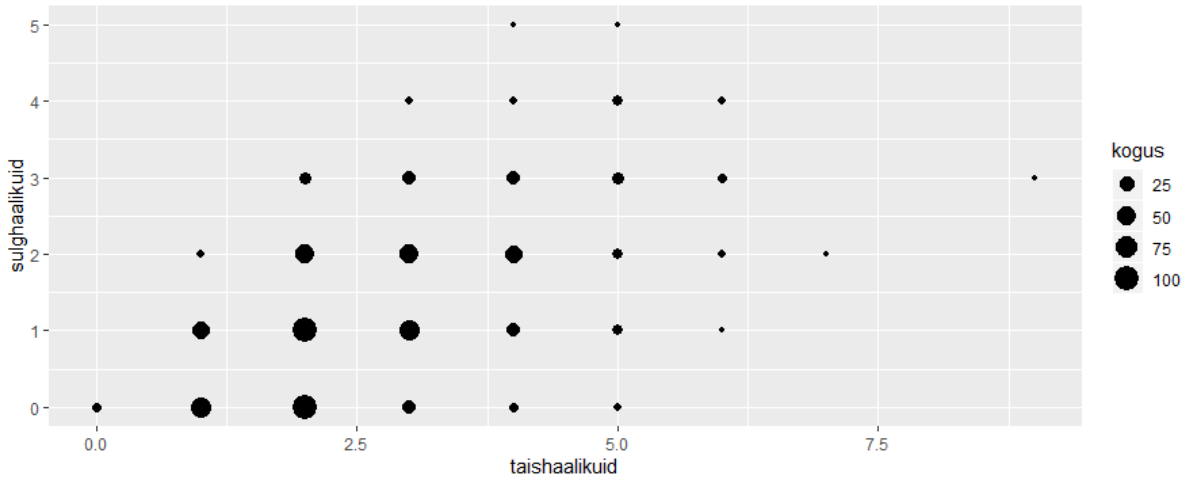
Mõnevõrra selgem isegi on ehk kokku sattuvate väärtuste näitamine suuruse järgi. Selleks arvutame välja, kui palju on iga täishääliku ja sulghääliku arvudega sõnu:

```
> sonad %>% group_by(taishaalikuid, sulghaalikuid) %>% summarise(kogus=n())
# A tibble: 31 x 3
# Groups:   taishaalikuid [9]
   taishaalikuid sulghaalikuid kogus
           <dbl>         <dbl> <int>
 1             0             0     6
 2             1             0    58
 3             1             1    40
 4             1             2     4
 5             2             0    98
 6             2             1   100
```

Paistab, et ilma täis- ja sulghäälikuteta sõnu on 6, ühe täishääliku ja puuduvate sulghäälikutega sõnu 58 jne.

Joonise koostamisel määrame juurde, et see kogus tuleb võtta punkti joonistamise suhteliseks suuruseks

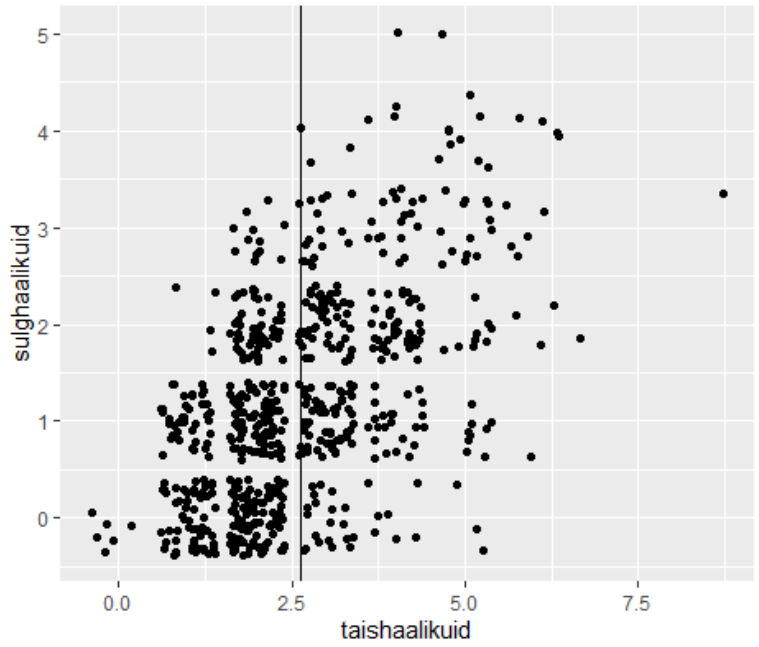
```
sonad %>% group_by(taishaalikuid, sulghaalikuid) %>% summarise(kogus=n()) %>%
  ggplot(aes(taishaalikuid, sulghaalikuid, size=kogus))+geom_point()
```



Abijooned

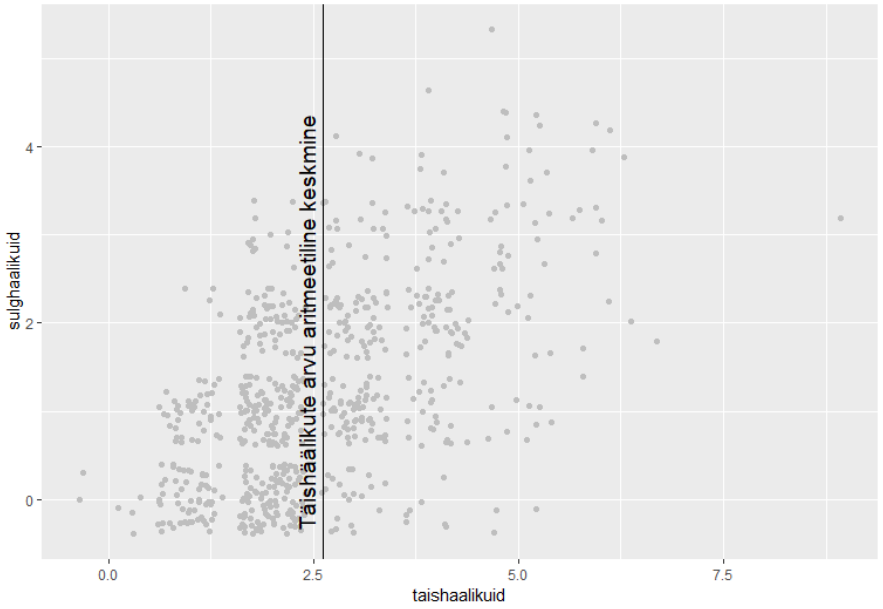
Lisaks andmete otsesele esitamisele aitavad jooniselt vajalikku välja lugeda täiendused. Lihtsamal juhul jooned mõne piiri tõmbamiseks. Siia vertikaalne joon täishäälikute arvu aritmeetilise keskmise juurde

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter()+
   geom_vline(xintercept=mean(sonad$taishaalikuid))
```



Joonisele ja joone peale kannatab ka eraldi teksti lisada. Lihtsamaks mooduseks käsklus annotate. Tüübiks text, kaasa koordinaadid ja sildi sisu. Teksti püstiseks saamiseks keerame seda 90 kraadi, vjust koos negatiivse väärtusega määrab, et tekst tuleks joone pealt veidi kõrgemale.

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+
  geom_jitter(color="gray")+
  geom_vline(xintercept=mean(sonad$taishaalikuid))+
  annotate('text', x=mean(sonad$taishaalikuid), y=2,
           label="Täishäälikute arvu aritmeetiline keskmine",
            size=5, angle=90, vjust=-0.5)
```



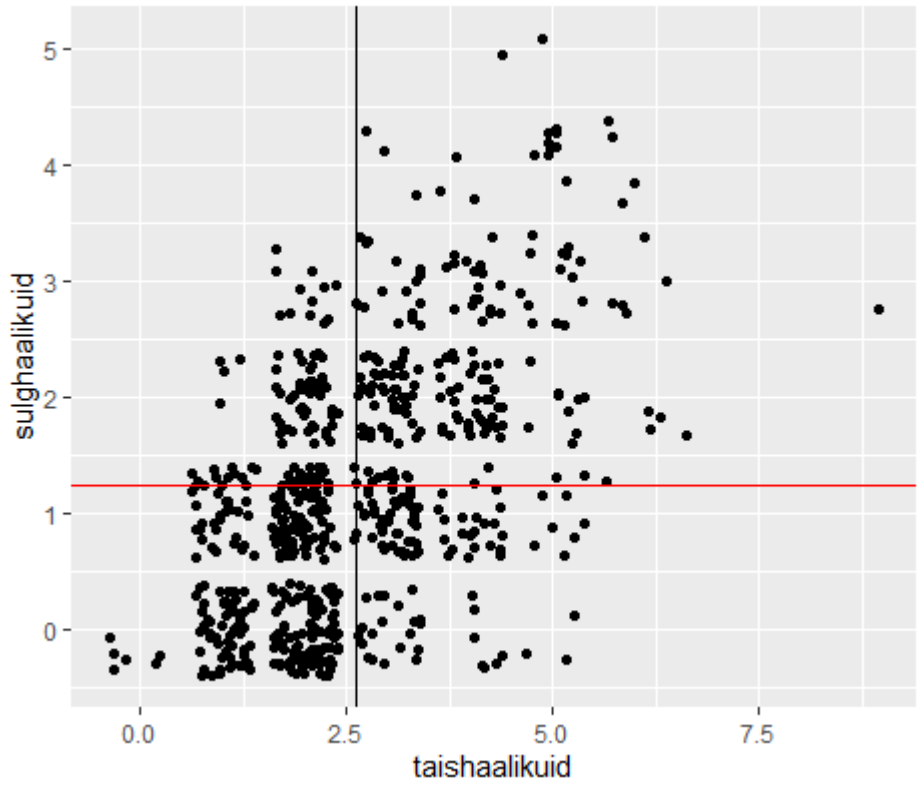
Sama tulemus on võimalik saada ka geom_text-käsu abil. Kasutada on aga seda mõistlik pigem siis, kui lisada on vaja tabelist suurem kogus andmeid. Parameetri data alt tuleb sisse uus tabel vajalike väärtustega - praegu ainult üks andmerida ühe x-i ja y-ga.

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+
  geom_jitter(color="gray")+
  geom_vline(xintercept=mean(sonad$taishaalikuid))+
    geom_text(data=tibble(x=mean(sonad$taishaalikuid), y=2,
                          t="Täishäälikute arvu aritmeetiline keskmine"),
              aes(x=x, y=y, label=t), size=5, angle=90, vjust=-0.5)
```

Harjutus

- Pange näited tööle
- Lisage horisontaaljoon sulghäälikute keskmise arvuga
- Värvige joon punaseks
- Kirjutage joonele selgitus

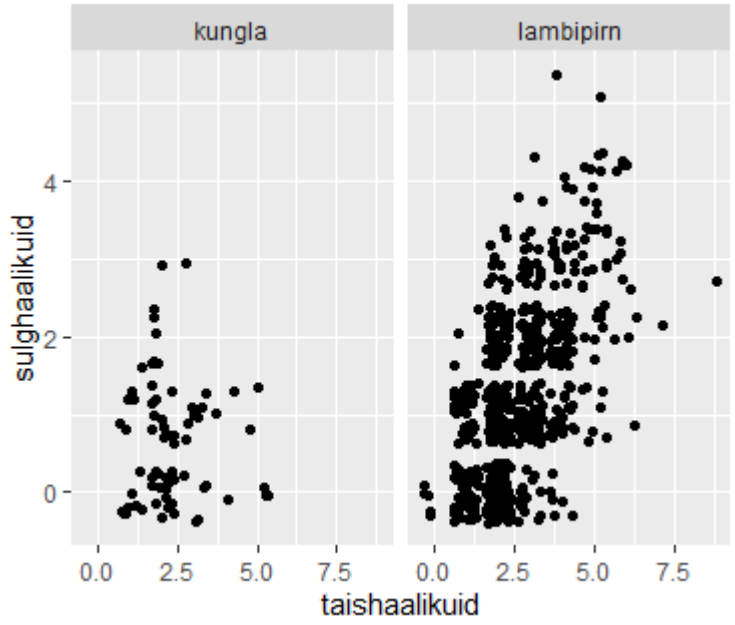
```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter()+
  geom_vline(xintercept=mean(sonad$taishaalikuid))+
  geom_hline(yintercept=mean(sonad$sulghaalikuid), color="red")
```



Mitu väiksemat joonist, facet_wrap

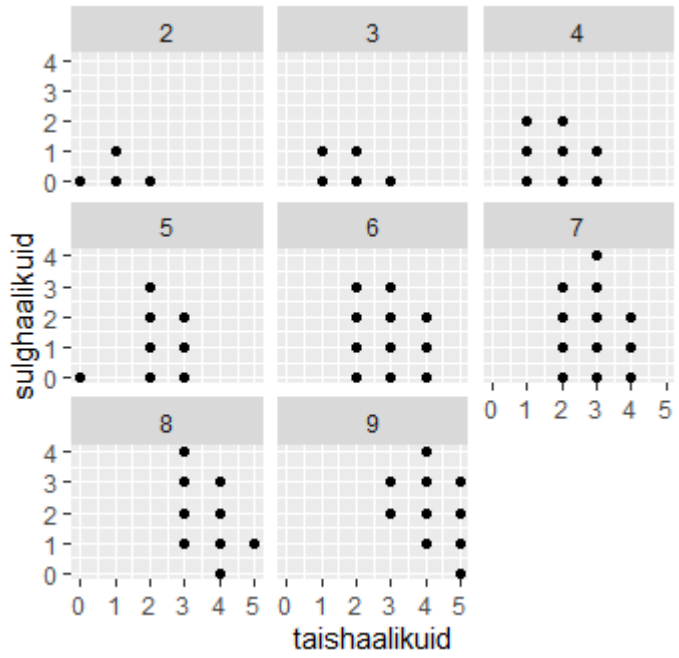
Tulemused lugude kaupa laiali jagada aitab facet_wrap, parameetrina tilde ~ ehk "õ konks" ning sellele järgnev jaotamistunnus - praegusel juhul lugu. Nii kummagi loo kohta väiksem joonis.

```
sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter()+
  facet_wrap(~lugu)
```



Siin aga parameetriks hoopis tähtede arv sõnas - ning iga arvu kohta eraldi joonis. Hõredate lõpunäidete eemaldamiseks jäetud vaid kümnest tähest lühemad sõnad

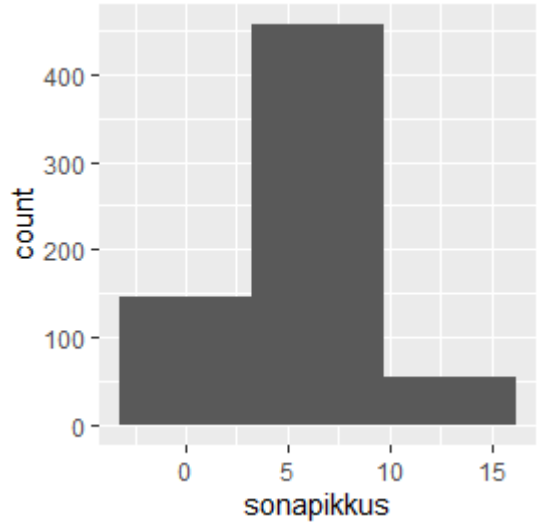
```
sonad %>% filter(sonapikkus<10) %>%
  ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point()+
  facet_wrap(~sonapikkus)
```



Eri joonised kahe tunnuse järgi, facet_grid

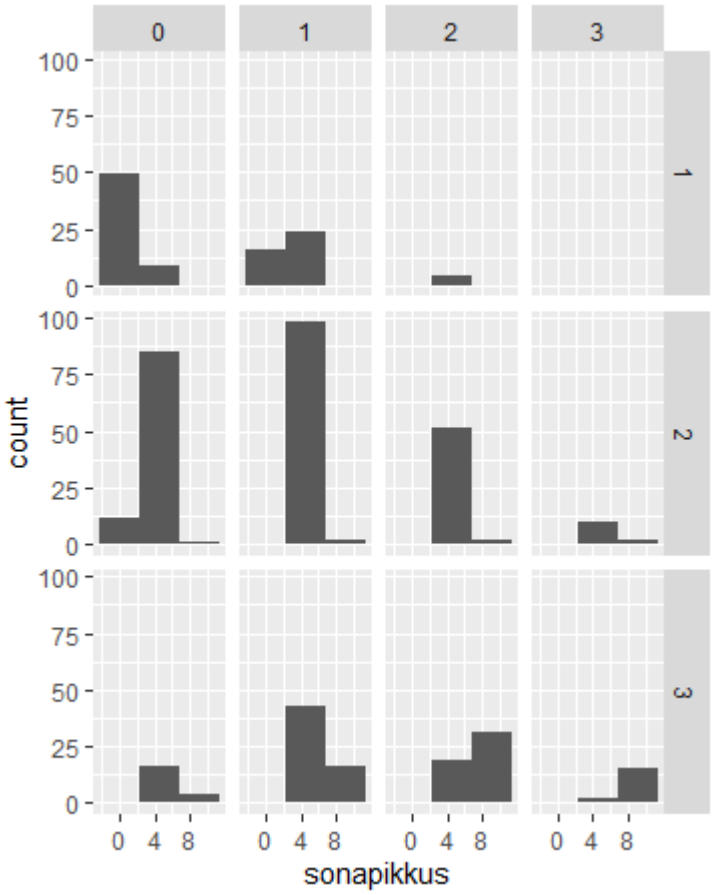
Näide histogrammi koostamise kohta.

```
sonad %>% filter(taishaalikuid <=5, sulghaalikuid <= 5) %>%
    ggplot(aes(sonapikkus)) + geom_histogram(bins = 3)
```



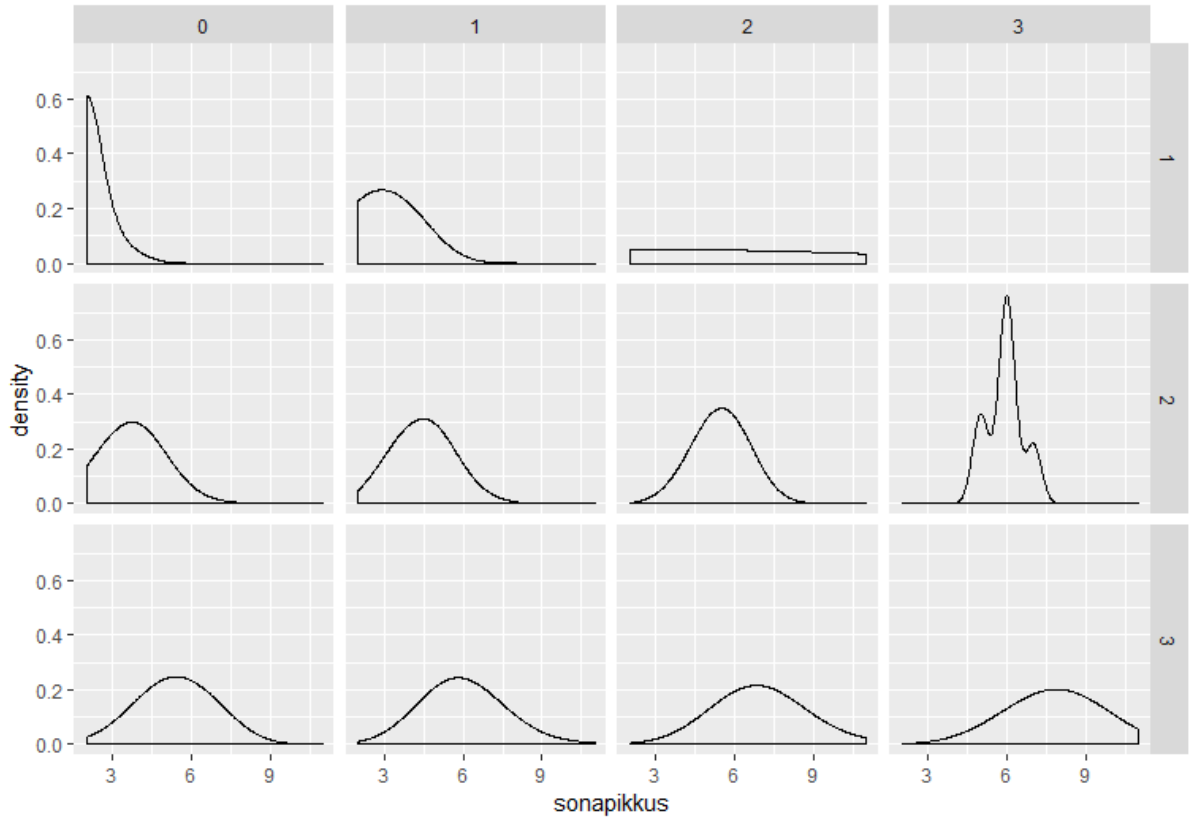
Edasi joonised ridadesse ja veergudesse - täishäälikuid ühest kolmeni, sulghäälikuid nullist kolmeni

```
sonad %>% filter(taishaalikuid>0, taishaalikuid <=3, sulghaalikuid <=3) %>%
    ggplot(aes(sonapikkus)) + geom_histogram(bins = 3) +
    facet_grid(taishaalikuid~sulghaalikuid)
```



Kui `geom_histogram` asendada käsuga `geom_density`, siis näidatakse tihedust joonena. Parameeter adjust näitab, et kuivõrd arvestatakse iga arvutamisel naaberpiirkondade andmeid - mida suurem arv, seda sujuvam joon.

```
sonad %>% filter(taishaalikuid>0, taishaalikuid <=3, sulghaalikuid <=3) %>%
  ggplot(aes(sonapikkus)) + geom_density(adjust=3) +
  facet_grid(taishaalikuid~sulghaalikuid)
```



ggplot-jooniste kohta mitmekülgsete näidetega varustatud tutvustuse leiab
`https://viz-ggplot2.rsquaredacademy.com/`

tidyverse-komplekti ja sealhulgas jooniste tuvustuse leiab

`https://r4ds.had.co.nz/data-visualisation.html`

Veel näiteid

`https://www.r-graph-gallery.com/index.html`

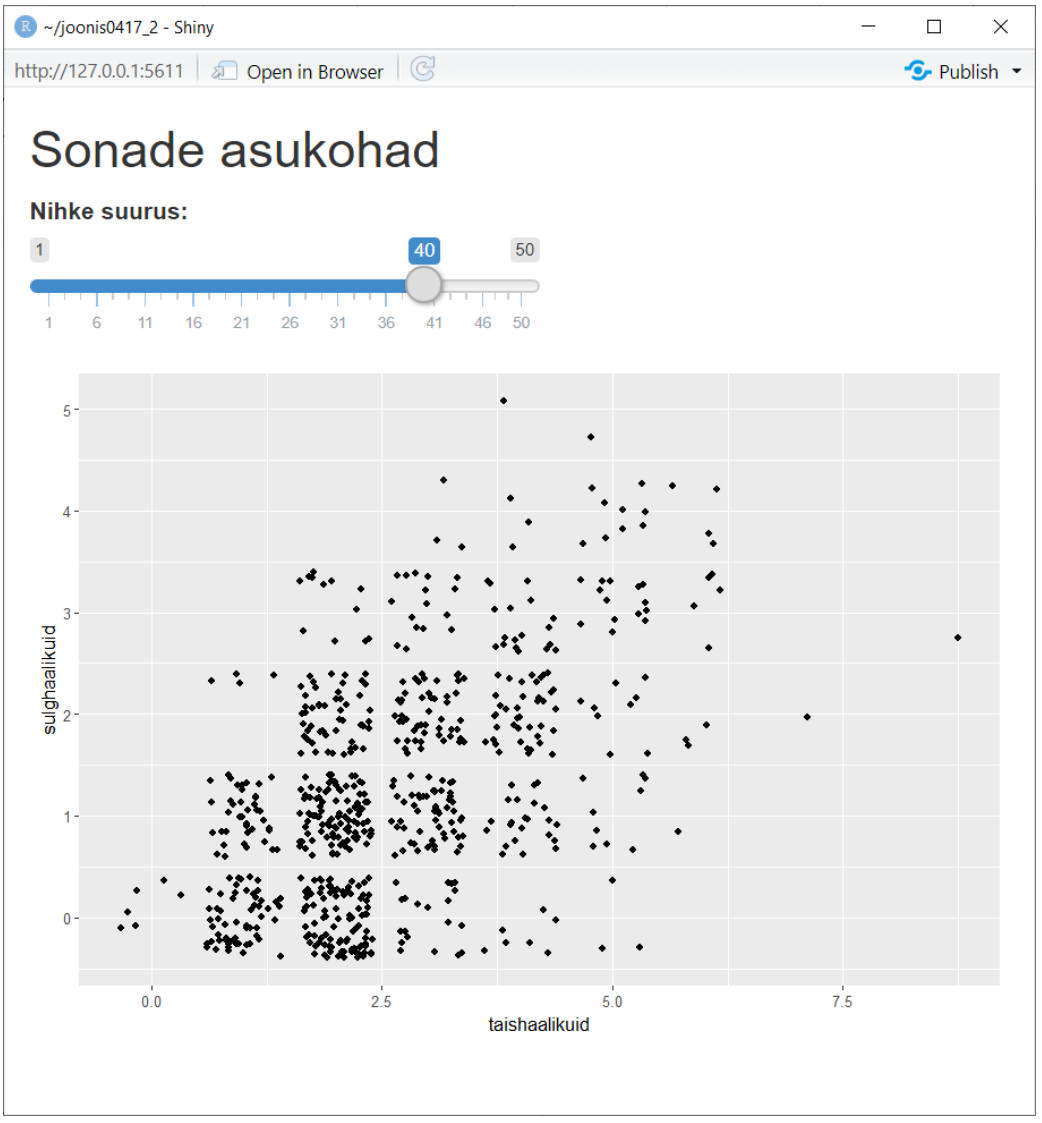
Shiny joonised

Shiny pakett võimaldab R-i abiga koostada interaktiivseid veebilehti. Kasutaja sisestab ja muudab andmeid ning nende põhjal kuvatakse vastused ning joonised. Installimiseks käsklus `install.packages('shiny')`. Edasi vaja sisse lugeda andmed, kujundada leht, luua serveripoolne arvutusosa ning lõpuks kogu rakendus käivitada shinyApp käsklusega.

```
library(shiny)
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
```

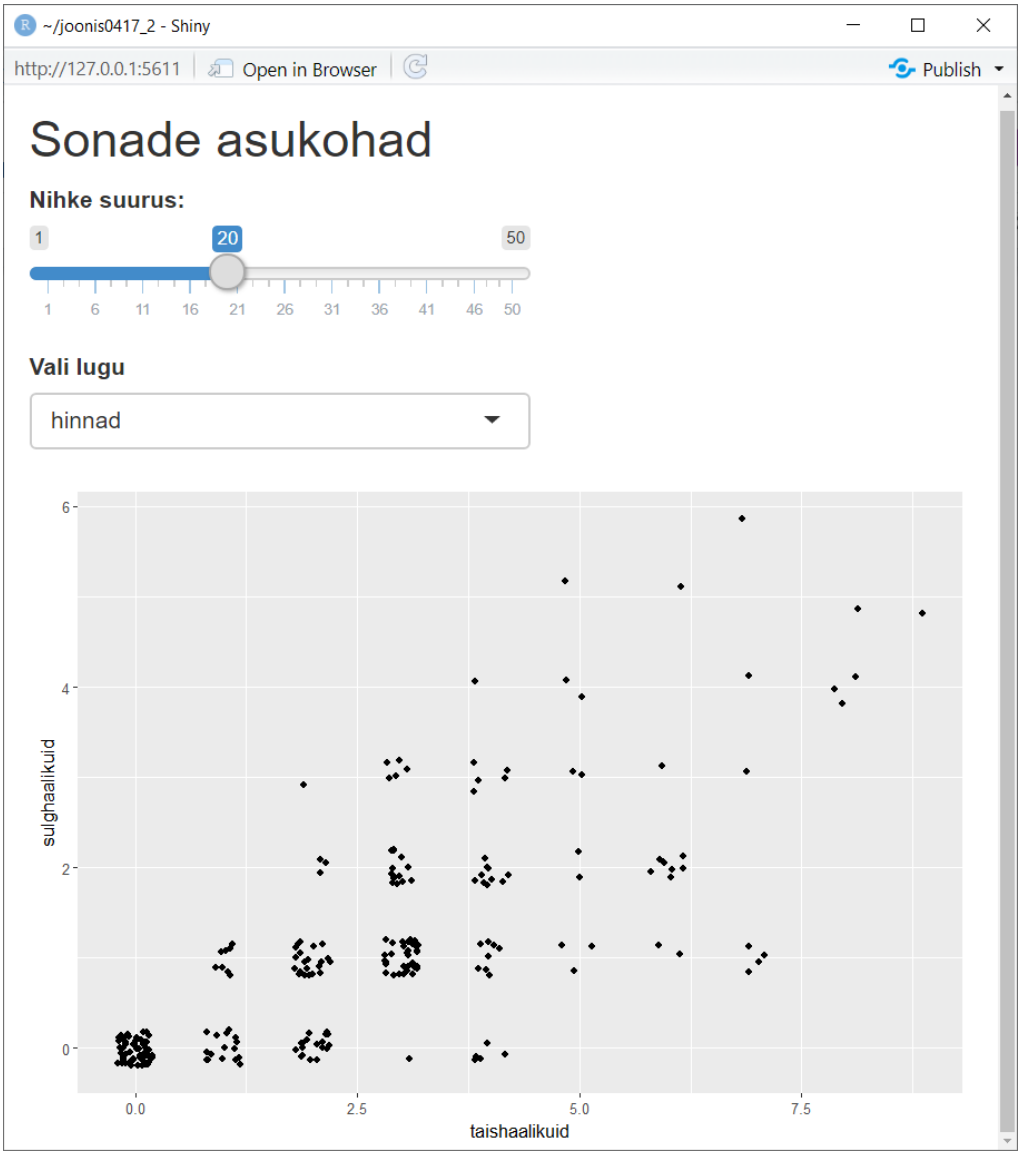
```
ui <- fluidPage(
    titlePanel("Sonade asukohad"),
    sliderInput("nihe", "Nihke suurus:", min = 1,  max = 50, value = 20),
    plotOutput("distPlot")
)

server <- function(input, output) {
    output$distPlot <- renderPlot({
        d=input$nihe/100.0
        sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter(width=d, height=d)
    })
}

shinyApp(ui = ui, server = server)
```

Näites saab kasutaja määrata joonisepunktide juhusliku nihke suurust. Koht distPlot paigutatakse kujunduse juures sobivasse kohta, serveriosas sellenimelisele väljundmuutujale omistatud joonis kuvatakse kujunduslehel.



Järgnevas näites lisatakse loo valik - et millise loo andmete põhjal joonis koostatakse. Valiku "koos" puhul jäetakse filtreerimata, ehk siis paistab tulemus kõigi andmete põhjal.

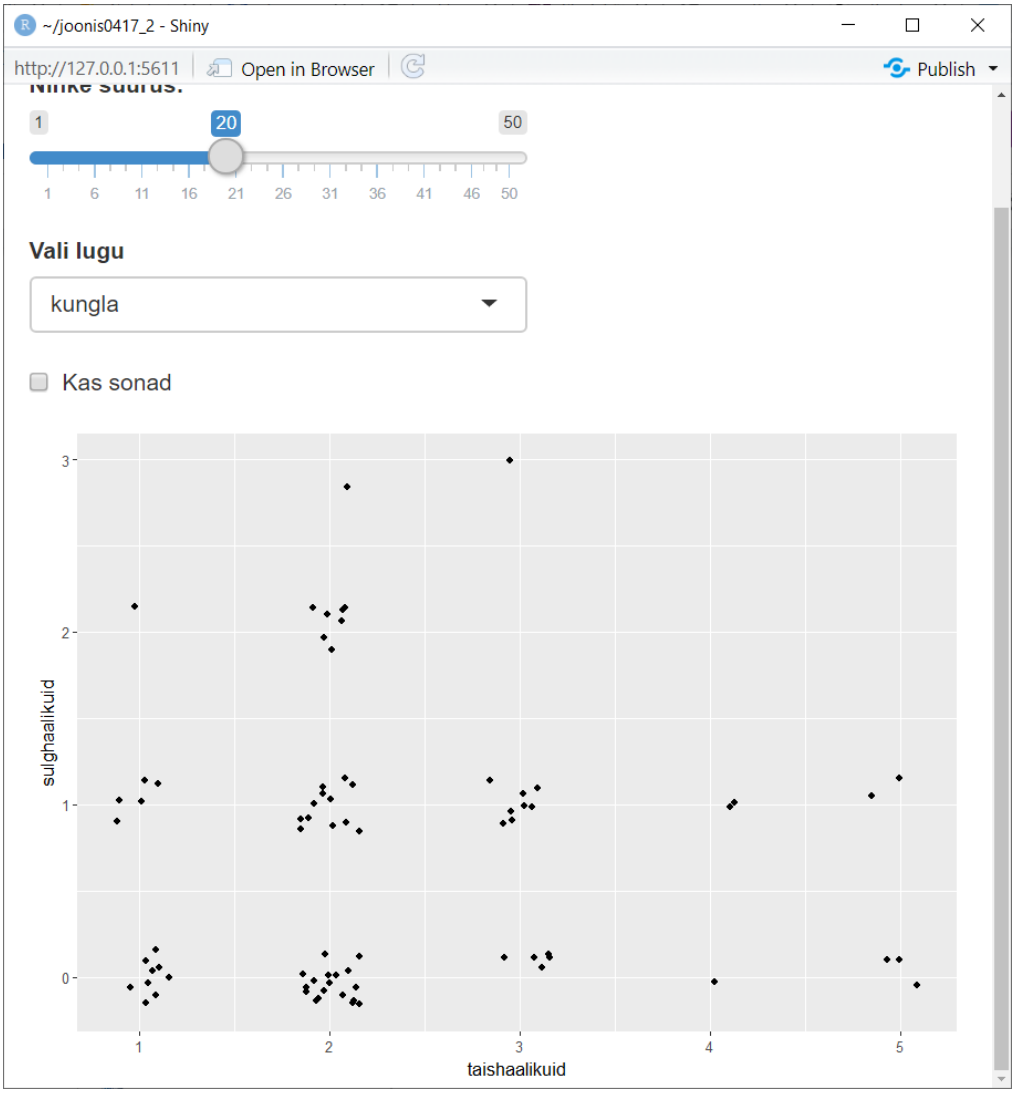
```
library(shiny)
library(tidyverse)
#sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haal
ikud.txt")
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikkuse
d_haalikud.txt")

ui <- fluidPage(
    titlePanel("Sonade asukohad"),
    sliderInput("nihe", "Nihke suurus:", min = 1,  max = 50, value = 20),
#    selectInput("lugu", "Vali lugu", c("kungla", "lambipirn", "molemad")),
    selectInput("lugu", "Vali lugu", c(unique(sonad$lugu), "koos")),
    plotOutput("distPlot")
)

server <- function(input, output) {
    output$distPlot <- renderPlot({
        d=input$nihe/100.0
        if(input$lugu=="koos"){
            s2=sonad
        } else {
          s2=sonad %>% filter(lugu==input$lugu)
        }
        s2 %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter(width=d, height=d)
    })
}

shinyApp(ui = ui, server = server)
```



Siinses näites juurde valik, et kas lisaks täpina kuvatud asukohtadele kuvatakse ka sõnu endid. Joonisel õnneks saab kihte rahumeeli plussmärgiga juurde lisada.

```
library(shiny)
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikkused_haalikud.txt")

ui <- fluidPage(
    titlePanel("Sonade asukohad"),
    sliderInput("nihe", "Nihke suurus:", min = 1,  max = 50, value = 20),
    selectInput("lugu", "Vali lugu", c(unique(sonad$lugu), "koos")),
    checkboxInput("kasSonad", "Kas sonad", TRUE),
    plotOutput("distPlot")
)

server <- function(input, output) {
    output$distPlot <- renderPlot({
        d=input$nihe/25.0
        if(input$lugu=="koos"){
```

```
            s2=sonad
        } else {
          s2=sonad %>% filter(lugu==input$lugu)
        }
        s2=s2 %>% mutate(taishaalikuid=jitter(taishaalikuid, factor=d),
sulghaalikuid=jitter(sulghaalikuid, d))
        joonis=s2 %>% ggplot(aes(taishaalikuid, sulghaalikuid))
        joonis=joonis+geom_point()
        if(input$kasSonad){
            joonis=joonis+geom_text(aes(label=sona))
        }
        joonis
    })
}

shinyApp(ui = ui, server = server)
```



Harjutus

- Tee näited läbi
- Lisa valik, kui pikki sõnu näidatakse

Animatsioonid

Liikuv pilt jääb mõnikord paremini silma ning aitab kiiremini ülevaate saada kui üksikute jooniste uurimine.

Pakett gganimate

Üheks mooduseks on ggplot-i täiendpaketi `gganimate` abil liikuvate piltide loomine. Vajadusel tuleb paketid enne installida

```
install.packages("gganimate")
install.packages("gifski")
install.packages("png")
```

ja andmed sisse lugeda

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

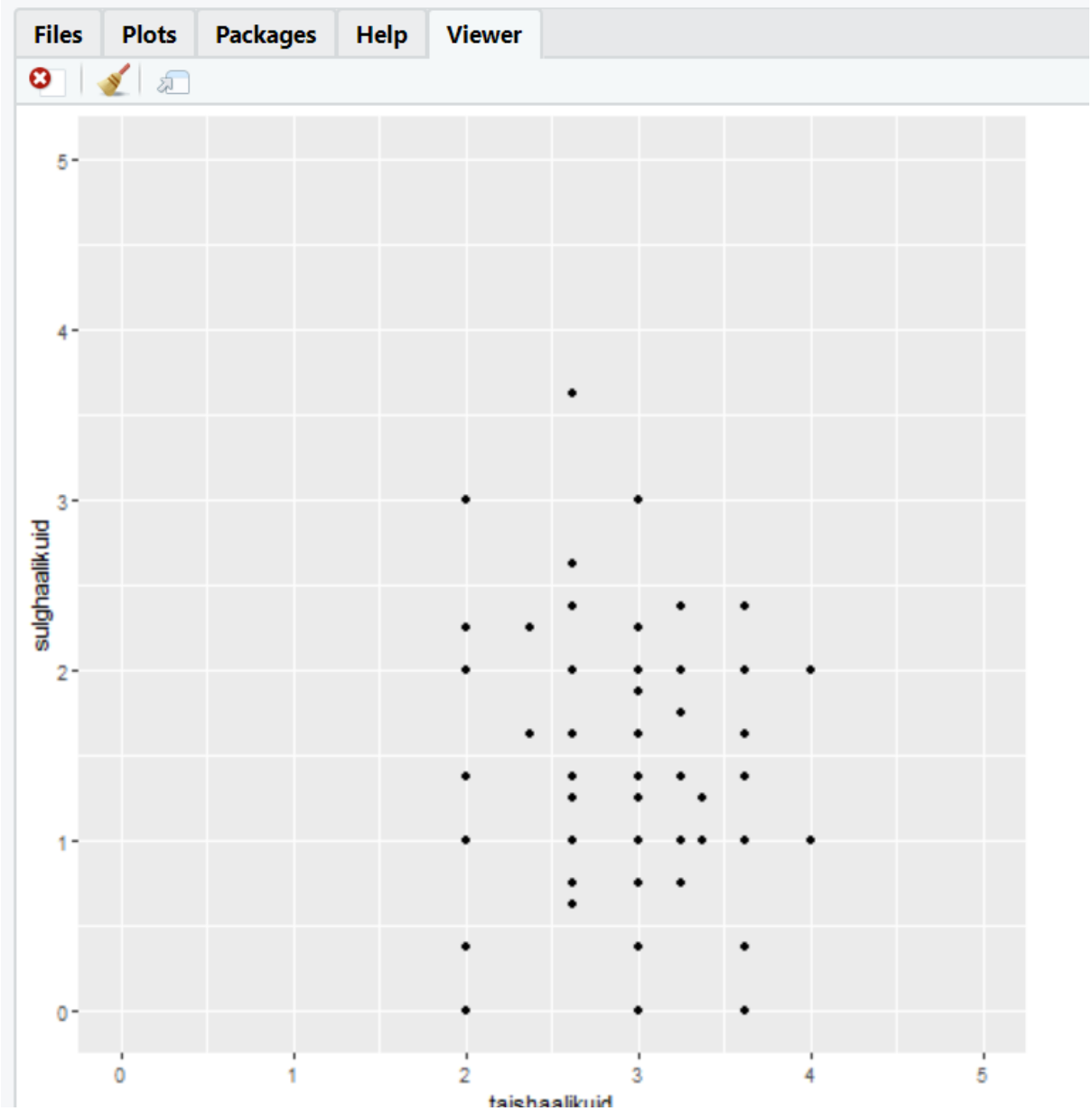
Joonise koostamisel saab lisada parameetri `transition_time`, määramaks, et millise parameetri muutmise kaudu järgmise kaadrini jõutakse. Video salvestamiseks/kuvamiseks hiljem käsklus `animate`.

```
library(gganimate)
library(gifski)
library(png)

pildid <- sonad %>% filter(taishaalikuid<6) %>%
   ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point()+
   transition_time(sonapikkus)
pildid %>% animate(renderer=gifski_renderer())
```

Käivitmise ajal näidatakse, kui kaugele arvutamisel jõuti

```
> pildid <- sonad %>% filter(taishaalikuid<6) %>%
+     ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point()+
+     transition_time(sonapikkus)
> pildid %>% animate(renderer=gifski_renderer())
Rendering [==============================>------------] at 8.9 fps ~ eta:  3s
```

Valmis animatsiooni näeb Vieweri aknast, sealt saab selle ka kettale salvestada



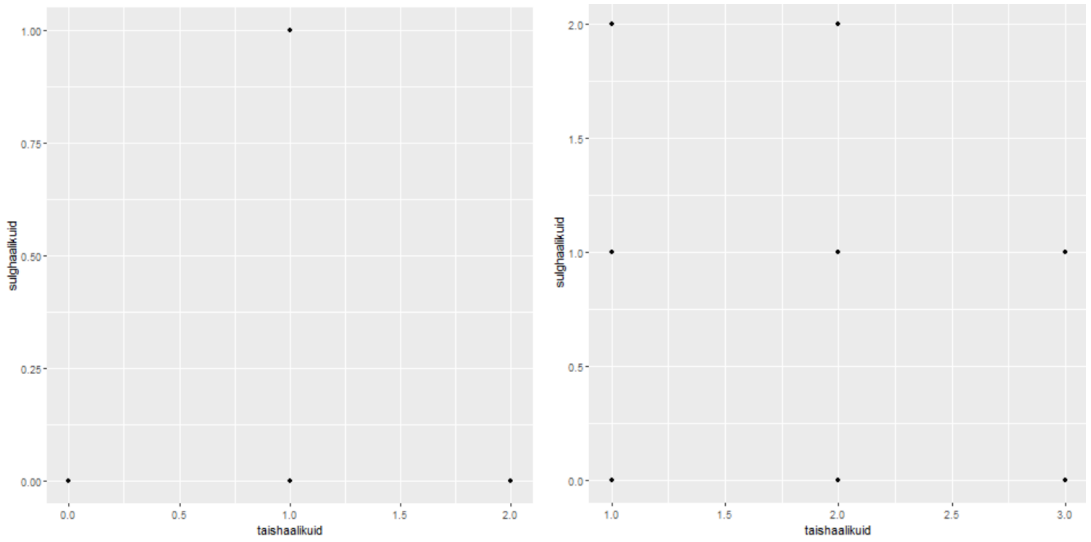
http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_gganimate.gif

Kui `gganimate` toimib, siis on tegemist mugava vahendiga ning mõnedki animatsioonid tehakse suhteliselt sujuvaks. Samas on sellega vahel raskusi installimisel ning kui mõne joonise vahekaadrite välja mõtlemisel hätta jäädakse, siis jääb tulemuseks sootuks ilma.

Pakett animation

Veidi puisem, kuid töökindlam on teek `animation`. Seal piisab käskluse saveGIF sisse jooniste printimisest ning heal juhul võibki tulemust vaadata. Siin koostaktakse animatsioon kahest pildist - ühel näha kahe ning teisel nelja tähega sõnade täis- ja sulghäälikute arvud.

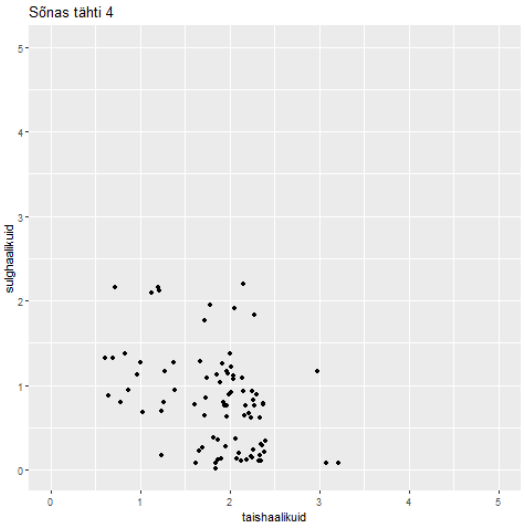
```
library(animation)
saveGIF({
  print(sonad %>% filter(sonapikkus==2) %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point())
  print(sonad %>% filter(sonapikkus==4) %>% ggplot(aes(taishaalikuid, sulghaalikuid))+geom_point())
})
```



http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_2pilti.gif

Lisaks võib määrata, et kuhu tulemus salvestatakse. Rohkemate piltide puhul on mugav need tsükli abil teha - `for`-tsükli puhul määratakse muutuja (praegusel juhul `nr`) järjestikused väärtused ning tehakse neist siis igaühega pilt. Skaala ka `xlim` ja `ylim` abil paika - siis püsivad järjestikused joonised mõõtkavas võrreldavad. Ning pealkirja juures näha, et mitme tähega joonis parajasti ees

```
saveGIF({
 for(nr in 0:6){
  print(sonad %>% filter(sonapikkus==nr) %>%
    ggplot(aes(taishaalikuid, sulghaalikuid))+geom_jitter()+
        xlim(0, 5) + ylim(0, 5)+ggtitle(paste("Sõnas tähti", nr)))
 }
}, movie.name="d:/animatsioon1.gif")
```



http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_haalikuid.gif

Mõned täiendused: 0 ja 1 tähega sõnu polnud. Iga täis- ja sulghäälikute paari kohta loetakse kokku, mitu sellist kombinatsiooni oli - ja vastavalt kuvatakse punkti suurus. Legend eemaldati, sest selle kuju muutus kippus ka pilti paigast ära ajama. Lõpus interval=2 näitab, et iga pilti näidatakse kaks sekundit.

```
saveGIF({
 for(nr in 2:10){
  print(sonad %>% filter(sonapikkus==nr) %>% group_by(taishaalikuid, sulghaalikuid)
         %>% summarise(kogus=n())
         %>% ggplot(aes(taishaalikuid, sulghaalikuid, size=kogus))+geom_point()+
           theme(legend.position="none")+
        xlim(0, 5) + ylim(0, 5)+ggtitle(paste("Sõnas tähti", nr)))
 }
}, movie.name="d:/animatsioon1.gif", interval=2)
```

http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_haalikuid_2.gif

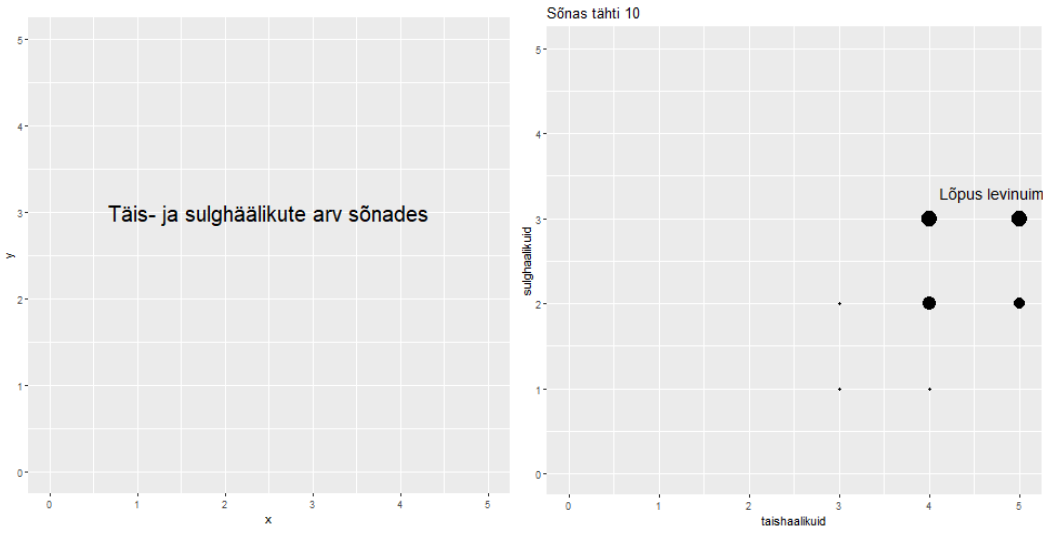
Sissejuhatav tekst

Käsu annotate abil saab joonisele lisada üksikuid kujundeid ja punkte, ka teksti. Kui sama kaadrit soovitakse pikemalt näidata, siis tuleb see mitu korda panna. Siin ongi sissejuhatav teksti kolmel kaadril tühjal joonisel.

Viimasel kolmel kaadril tuleb juurde tekst "lõpus levinuim" mummule, kus 10-täheliste sõnade juures enim sulg- ja täishääliku paare.

```
library(animation)
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
saveGIF({
  for(i in 1:3){
    print(ggplot()+xlim(0, 5)+ylim(0, 5)+
        annotate("text", 2.5, 3, label="Täis- ja sulghäälikute arv sõnades", size=7))
```

```
  }
  for(nr in 2:10){
    print(sonad %>% filter(sonapikkus==nr) %>% group_by(taishaalikuid, sulghaalikuid)
          %>% summarise(kogus=n())
          %>% ggplot(aes(taishaalikuid, sulghaalikuid, size=kogus))+geom_point()+
            theme(legend.position="none")+
            xlim(0, 5) + ylim(0, 5)+ggtitle(paste("Sõnas tähti", nr)))
  }
  nr=10
  for(i in 1:3){
   print(sonad %>% filter(sonapikkus==nr) %>% group_by(taishaalikuid, sulghaalikuid)
        %>% summarise(kogus=n())
        %>% ggplot(aes(taishaalikuid, sulghaalikuid, size=kogus))+geom_point()+
          theme(legend.position="none")+
          xlim(0, 5) + ylim(0, 5)+ggtitle(paste("Sõnas tähti", nr))+
          annotate("text", x=4.7, y=3.3, label="Lõpus levinuim", size=5))
  }
}, movie.name="d:/animatsioon1.gif", interval=2)
```



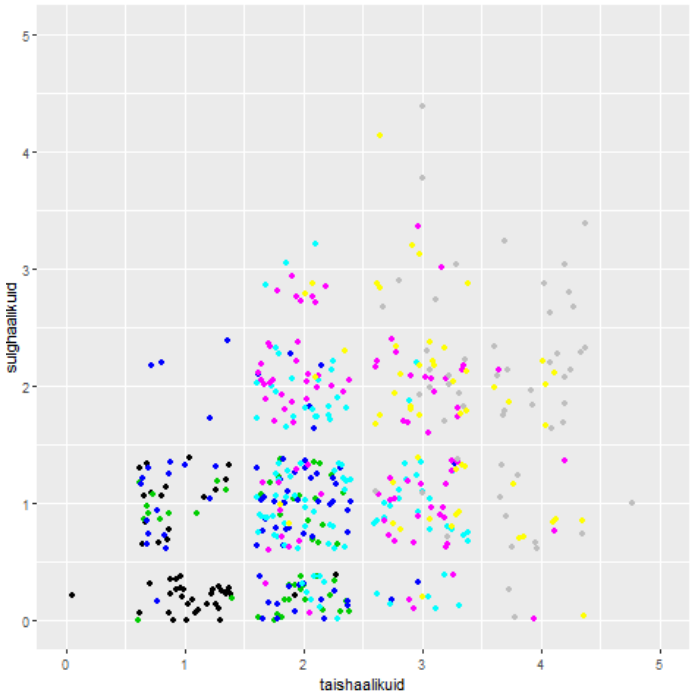
http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_pealkiri.gif

Joonise täiendamine

Paketi ggplot2 joonised kannatavad olemasolevale joonisele järjest kihte ehk näidatavaid väärtusi lisada. Siin näites pannakse iga pikkuse puhul juurde vastava tähtede arvuga sõnade asukohad täis- ja sulghäälikute joonisel. Värviks määratakse sõnapikkuse number, millele vastavalt R siis praegu ise värvi valib.

```
saveGIF({
  nr=2
  joonis <-
    sonad %>% filter(sonapikkus==nr) %>%
     ggplot(aes(taishaalikuid, sulghaalikuid))+
     geom_jitter()+ theme(legend.position="none")+
    xlim(0, 5) + ylim(0, 5)
  print(joonis)
  for(nr in 3:10){
    joonis <- joonis+geom_jitter(data=sonad %>% filter(sonapikkus==nr), col=nr)
```

```
    print(joonis)
  }
}, movie.name="d:/animatsioon1.gif", interval=2)
```

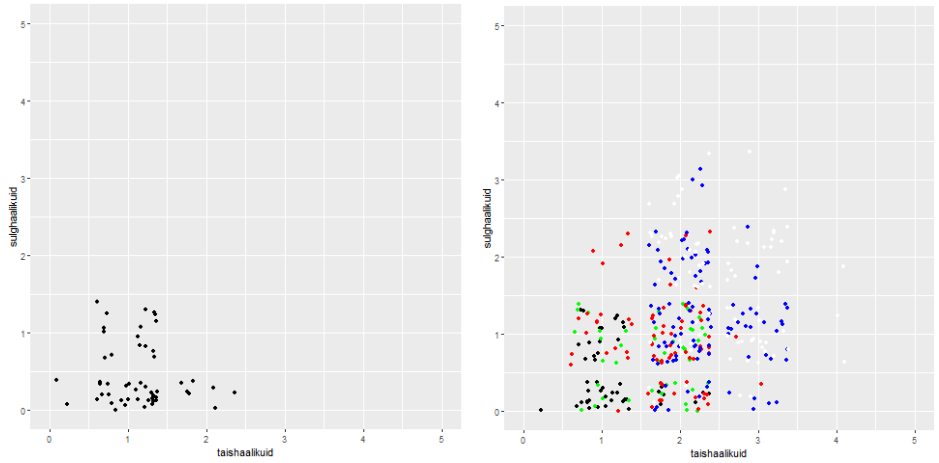


http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_taiend.gif

Omavalitud värvid

Eelmisega sarnane lahendus, ainult, et värvid käsitsi määratud. Esimesed kaks kohta tühjad, sest ühetähelised sõnad puuduvad ning kahetäheliste puhul joonistatakse algul mustad täpid, mis on vaikimisi värviks

```
varvid=c("", "", "green", "red", "blue", "white")
saveGIF({
  nr=2
  joonis <-
    sonad %>% filter(sonapikkus==nr) %>%
     ggplot(aes(taishaalikuid, sulghaalikuid))+
     geom_jitter()+ theme(legend.position="none")+
    xlim(0, 5) + ylim(0, 5)
  print(joonis)
  for(nr in 3:6){
    joonis <- joonis+
       geom_jitter(data=sonad %>% filter(sonapikkus==nr), col=varvid[nr])
    print(joonis)
  }
}, movie.name="d:/animatsioon1.gif", interval=2)
```

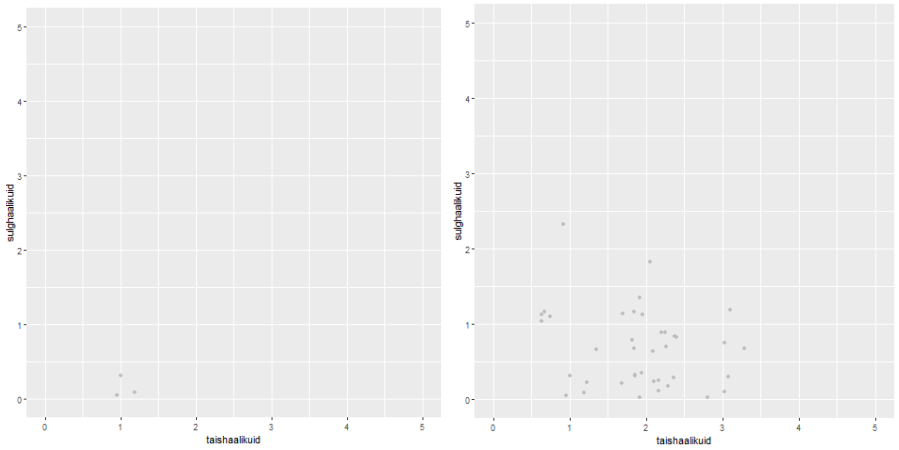


http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_varvid.gif

Ettearvutatud kohtadega punktid

Käsklus geom_jitter loksutab punktid sobiva koha lähedale joonistamise käigus. Nõnda aga korduval joonistamisel sõnade andmeid lisades hüppavad ka algsed punktid paigast ära. Kui soovime, et punktid poleks üksteise peal, samas aga püsiksid järgnevates kaadrites paigal, siis tasub uued koordinaadid eelnevalt välja arvutada. Käsk mutate loob uue tulba - või siis arvutab vanale uue väärtuse, jitter liigutab väärtused algse väärtuse lähedale - parameeter factor=2 näitab, et võrreldes tavalisega võib kaks korda rohkem kaugemale ajada - nii saab rohkem ruumi kaetud ning ei satuta niivõrd kohakuti. Lisaks arvutatakse sõnadele juurde reanumber, nii võimalik loo sõnade juures ilusasti järjest liikuda. Tsüklis joonistamine kuni kaadrile vastava pikkusega sõnani.

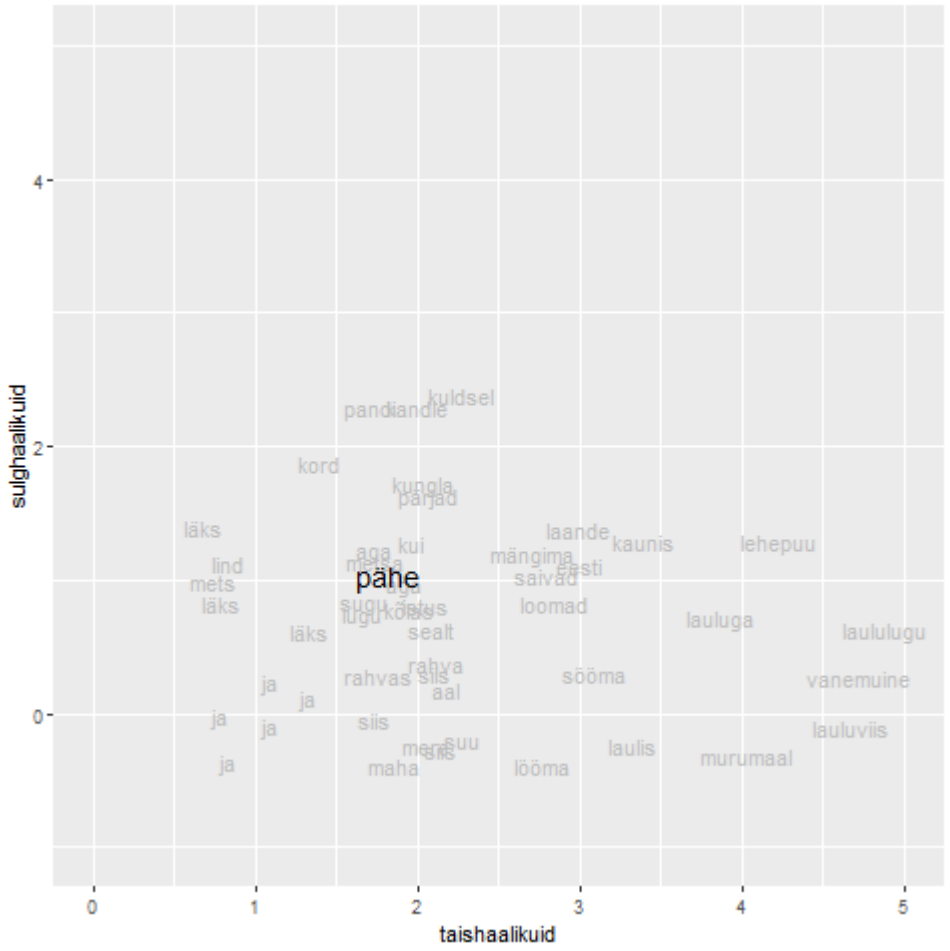
```
s2 <- sonad %>% filter(lugu=="kungla") %>% arrange(sonapikkus) %>%
                mutate(taishaalikuid=jitter(taishaalikuid, factor=2),
                       sulghaalikuid=jitter(sulghaalikuid, factor=2),
                       snr=row_number())
saveGIF({
  for(nr in 2:nrow(s2)){
    print(s2 %>% filter(snr<=nr) %>%
       ggplot(aes(taishaalikuid, sulghaalikuid)) + geom_point(color="gray")+
       xlim(0, 5) + ylim(0, 5))
  }
}, movie.name="d:/animatsioon2.gif", interval=0.1)
```



http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_tapid_sonapikkus.gif

Viimases animatsiooninäites on algul tutvustav tekst, iga uus lisanduv sõna musta värvi, teised sõnad väiksemad ja hallid.

```
s2 <- sonad %>% filter(lugu=="kungla")  %>%
                mutate(taishaalikuid=jitter(taishaalikuid, factor=2),
                       sulghaalikuid=jitter(sulghaalikuid, factor=2),
                       snr=row_number())
saveGIF({
  for(i in 1:2){
    print(ggplot()+xlim(0, 5)+ylim(-1, 5)+
            annotate("text", 2.5, 3, label="Täis- ja sulghäälikute arv sõnades", size=7))
  }
  print(ggplot()+xlim(0, 5)+ylim(-1, 5))
  for(nr in 1:nrow(s2)){
    print(s2 %>% filter(snr<nr) %>%
       ggplot(aes(taishaalikuid, sulghaalikuid, label=sona)) + geom_text(color="gray")+
       xlim(0, 5) + ylim(-1, 5)+
        geom_text(data=s2 %>% filter(snr==nr), color="black", size=5))
  }
}, movie.name="d:/animatsioon2.gif", interval=1)
```



http://www.tlu.ee/~jaagup/dh/kvantitatiivne_digihumanitaaria/anim_sonad_kungla.gif

Tabel laiemale kujule

Andmeid haarata on lihtsam, kui need silme ette korraga ära mahuvad. Sellise olukorra tekitamiseks näide.

Kõigepealt koostame sagedusloendi sõnapikkuste kaupa:

```
> sonad %>% group_by(sonapikkus) %>% summarise(kogus=n())
# A tibble: 15 x 2
   sonapikkus kogus
        <int> <int>
 1          2    82
 2          3    64
 3          4    98
 4          5   112
 5          6    92
 6          7    61
 7          8    54
 8          9    40
 9         10    26
10         11    13
11         12    15
```

```
12          13      7
13          14      4
14          15      3
15          19      1
```

Järgmise sammuna näitame kummagi loo sõnade pikkusi eraldi. Kõigepealt rühmitame read group_by abil ning pärast tehteid eemaldame rühmituse ungroup-iga

```
pikkused <-
  sonad %>% group_by(lugu, sonapikkus) %>%
  summarise(kogus=n()) %>% ungroup()

head(pikkused, 10)
```

```
   lugu      sonapikkus kogus
   <chr>          <int> <int>
 1 kungla             2     9
 2 kungla             3     8
 3 kungla             4    21
 4 kungla             5    12
 5 kungla             6    14
 6 kungla             7     5
 7 kungla             8     2
 8 kungla             9     4
 9 lambipirn          2    73
10 lambipirn          3    56
```

Käsklus spread muudab tulemuse laiale kujule. Parameetrina sõnapikkuse väärtustest saavad tulpade nimed ning kogusest tuleb tabeli lahtrite sisu. Tulba "lugu" eraldi väärtustest saavad uue loodava tabeli read.

```
> spread(pikkused, sonapikkus, kogus)
# A tibble: 2 x 16
  lugu   `2`   `3`   `4`   `5`   `6`   `7`
  <ch> <int> <int> <int> <int> <int> <int>
1 kun~     9     8    21    12    14     5
2 lam~    73    56    77   100    78    56
# ... with 9 more variables: `8` <int>,
```

Mugavama vaatamiskuju annab käsklus View. Puuduvate andmete kohale nullide kirjutamise määrab parameeter fill=0, Kungla rahva juures lihtsalt pole kümnest tähest pikemaid sõnu.

```
> View(spread(pikkused, sonapikkus, kogus, fill=0))
```

	lugu	2	3	4	5	6	7	8	9	10	11	12	13	14	15	19
1	kungla	9	8	21	12	14	5	2	4	0	0	0	0	0	0	0
2	lambipirn	73	56	77	100	78	56	52	36	26	13	15	7	4	3	1

Harjutus

- Jäta sõnade andmestikust alles ainult kolmetähelised sõnad
  http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt
- Loe Kungla rahva loos kokku, mitu korda milline kolmetäheline sõna esineb
- Tee sama ka lambipirni loo puhul
- Loo nende põhjal tabel, kus tulpadeks on loo nimi, sõna ning sõna esinemiste arv loos. (kui vastav sõna loos puudub, siis nulliga ridu ei kuvata)
- Loo nende põhjal lai tabel, kus tulpadeks on loo nimi ning iga erinev kolmetäheline sõna. Kuva tabelis vastavate sõnade esinemise arv loos.

- Koosta sõnade andmestikust sarnane lai tabel ühe käsureaga group_by ja spread käskude abil

Lahendus

Kungla rahva erinevad kolmetähelised sõnad koos kogustega

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
sonad %>% filter(lugu=="kungla", sonapikkus==3) %>% group_by(sona) %>% summarise(kogus=n())

# A tibble: 7 x 2
  sona  kogus
  <chr> <int>
1 aal       1
2 aga       2
3 kui       1
4 mäe       1
5 sai       1
6 see       1
7 suu       1
```

Andmed laulude kaupa muutujatesse

```
kunglasonad<-sonad %>% filter(lugu=="kungla", sonapikkus==3) %>% group_by(sona) %>%
summarise(kogus=n())
lambisonad<-sonad %>% filter(lugu=="lambipirn", sonapikkus==3) %>% group_by(sona) %>%
summarise(kogus=n())


> lambisonad
# A tibble: 30 x 2
   sona  kogus
   <chr> <int>
 1 aga       5
 2 all       1
 3 aru       1
 4 asi       2
 5 ehk       1
 6 ise       1
 7 jne       4
 8 kas       1
 9 kes       5
10 kui       3
# ... with 20 more rows
```

Kahe tabeli ühendamine

```
> kunglasonad %>% bind_rows(lambisonad)
# A tibble: 37 x 2
   sona  kogus
   <chr> <int>
 1 aal       1
 2 aga       2
 3 kui       1
 4 mäe       1
 5 sai       1
 6 see       1
 7 suu       1
 8 aga       5
 9 all       1
10 aru       1
# ... with 27 more rows
```

Ühendamisel kasulik teada, kummast tabelist andmed tulid

```
> kunglasonad %>% mutate(lugu="kungla")
# A tibble: 7 x 3
  sona  kogus lugu
  <chr> <int> <chr>
1 aal       1 kungla
2 aga       2 kungla
3 kui       1 kungla
4 mäe       1 kungla
5 sai       1 kungla
6 see       1 kungla
7 suu       1 kungla
```

Kui tahta lisatud tulp tõsta ette, siis aitab käsklus select soovitud esimese tulbaga ning lõppu `everything()`, mis kuvab kõik ülejäänud tulbad

```
> kunglasonad %>% mutate(lugu="kungla") %>% select(lugu, everything())
# A tibble: 7 x 3
  lugu   sona  kogus
  <chr>  <chr> <int>
1 kungla aal       1
2 kungla aga       2
3 kungla kui       1
4 kungla mäe       1
5 kungla sai       1
6 kungla see       1
7 kungla suu       1
```

Mõlema tabeli andmed koos lugude nimedega

```
> kunglasonad %>% mutate(lugu="kungla") %>% bind_rows(lambisonad %>%
mutate(lugu="lambipirn"))
# A tibble: 37 x 3
   sona  kogus lugu
   <chr> <int> <chr>
 1 aal       1 kungla
 2 aga       2 kungla
 3 kui       1 kungla
 4 mäe       1 kungla
 5 sai       1 kungla
 6 see       1 kungla
```

```
 7 suu       1 kungla
 8 aga       5 lambipirn
 9 all       1 lambipirn
10 aru       1 lambipirn
# ... with 27 more rows
```

Juurde laiale kujule viimise käsk

```
> kunglasonad %>% mutate(lugu="kungla") %>% bind_rows(lambisonad %>%
mutate(lugu="lambipirn")) %>%  spread(sona, kogus, fill=0)
# A tibble: 2 x 34
  lugu     aal   aga   all   aru   asi   ehk   ise   jne   kas   kes   kui   loo   mis
mul   mäe
  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
<dbl> <dbl>
1 kungla     1     2     0     0     0     0     0     0     0     0     1     0     0
0     1
2 lambi~     0     5     1     1     2     1     1     4     1     5     3     1     1
1     0
# ... with 18 more variables: nad <dbl>, nii <dbl>, ole <dbl>, oma <dbl>, pea <dbl>, peo
<dbl>,
#   saa <dbl>, sai <dbl>, see <dbl>, suu <dbl>, tal <dbl>, usu <dbl>, uue <dbl>, või <dbl>,
#   õde <dbl>, ära <dbl>, ühe <dbl>, üks <dbl>
```

Eri sõnade arv lugude kaupa kahe tulba järgi rühmitades

```
> sonad %>% filter(sonapikkus==3) %>% group_by(lugu, sona) %>% summarise(kogus=n())
# A tibble: 37 x 3
# Groups:   lugu [2]
   lugu      sona  kogus
   <chr>     <chr> <int>
 1 kungla    aal       1
 2 kungla    aga       2
 3 kungla    kui       1
 4 kungla    mäe       1
 5 kungla    sai       1
 6 kungla    see       1
 7 kungla    suu       1
 8 lambipirn aga       5
 9 lambipirn all       1
10 lambipirn aru       1
# ... with 27 more rows
```

Sama koos laiaks tabeliks tegemisega

```
> sonad %>% filter(sonapikkus==3) %>% group_by(lugu, sona) %>% summarise(kogus=n()) %>%
spread(sona, kogus, fill=0)
```

Vaade View abil

```
> View(sonad %>% filter(sonapikkus==3) %>% group_by(lugu, sona) %>% summarise(kogus=n())
%>% spread(sona, kogus, fill=0))
```

lugu	aal	aga	all	aru	asi	ehk	ise	jne	kas	kes	kui	loo	mis
kungla	1	2	0	0	0	0	0	0	0	0	1	0	0
lambipirn	0	5	1	1	2	1	1	4	1	5	3	1	1

Üldistamine, proportsioonide test

Mõõtmisel saadakse kätte hulk tulemusi, mis näitavad arve otseselt mõõdetud objektide kohta ning need sobivad kindlasti nende samade objektide ja seisu kirjeldamiseks. Kuivõrd on võimalik kätte saadud tulemusi üldistada laiemaks kasutamiseks, see on juba keerulisem ning selle tarbeks on andmeanalüüsivaldkonnas kokku pandud hulk teste. R-keeles lihtsamaks neist on prop.test ehk proportsioonide test.

Ettevalmistus

Pakett mällu ning andmed muutujasse.

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Andmestikust juhuslikult valitud kümne sõna andmed `sample_n` käsu abil

```
> sample_n(sonad, 10)
# A tibble: 10 x 5
   lugu      sona         sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>             <int>         <int>         <int>
 1 lambipirn hotellitoas          11             5             2
 2 lambipirn see                   3             2             0
 3 lambipirn ilmselt               7             2             1
 4 lambipirn võimalik              8             4             1
 5 lambipirn keerab                6             3             2
 6 lambipirn ka                    2             1             1
 7 kungla    ja                    2             1             0
 8 lambipirn sedakorda             9             4             3
 9 lambipirn on                    2             1             0
10 lambipirn kahe                  4             2             1
```

Uuel käivitamisel tulevad sootuks teised sõnad

```
> sample_n(sonad, 10)
# A tibble: 10 x 5
   lugu      sona            sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>                <int>         <int>         <int>
 1 kungla    rahvas                   6             2             0
 2 lambipirn kirurgi                  7             3             2
 3 lambipirn pirni                    5             2             1
 4 lambipirn funktsioneeriva         15             7             2
 5 lambipirn valgusallikata          14             6             3
 6 lambipirn joodud                   6             3             2
 7 lambipirn tõene                    5             3             1
 8 lambipirn toolile                  7             4             1
 9 lambipirn käigus                   6             3             2
```

```
10 kungla    siis                    4             2             0
```

Ja kolmandal korral hoopis midagi muud. Kusjuures paistab, et esimesel korral oli üks sõna kümnest Kungla rahva loost, teisel kaks ning kolmandal mitte ühtegi. Kuna kõigil kordadel valiti juhuslikud read, siis nii ühe, kahe kui ka kolme Kungla rahva sõna esinemine kümnest sõnast on täiesti loomulik, kui peaksime püüdma juhuslikult leitud sõnade põhjal püüdma hinnata sõnade sageduste suhet üldkogumis ehk mõlema täisteksti kogumis.

```
> sample_n(sonad, 10)
# A tibble: 10 x 5
   lugu      sona          sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>              <int>         <int>         <int>
 1 lambipirn ta                     2             1             1
 2 lambipirn ei                     2             2             0
 3 lambipirn siis                   4             2             0
 4 lambipirn ja                     2             1             0
 5 lambipirn ole                    3             2             0
 6 lambipirn politseinikud         13             6             4
 7 lambipirn traumapunkti          12             5             4
 8 lambipirn õrnalt                 6             2             1
 9 lambipirn sest                   4             1             1
10 lambipirn traumapunkti          12             5             4
```

Ainult arvude kätte saamiseks rühmitame sõnad loo järgi ja loemegi kohe esinemiskordade arvud. Nagu paistab, siis võib tulla ette ka olukord, kus mõlemaid on ühepalju

```
> sample_n(sonad, 10) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla        5
2 lambipirn     5
```

Kungla rahva omi on kümnendik

```
> sample_n(sonad, 10) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla        1
2 lambipirn     9
```

või puuduvad nad sootuks.

```
> sample_n(sonad, 10) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 1 x 2
  lugu      kogus
  <chr>     <int>
1 lambipirn    10
```

Arve saja kaupa küsides kõiguvad tulemused suhtarvuna veidi vähem, absoluutarvuna aga ikka märkimisväärselt : sajast 14, 10 ja 8 esinemiskorda.

```
> sample_n(sonad, 100) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla       14
2 lambipirn    86

> sample_n(sonad, 100) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla       10
2 lambipirn    90

> sample_n(sonad, 100) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla        8
2 lambipirn    92
```

Võib ka küsida sada juhuslikku sõna ning siis filtreerida loo järgi välja ja küsida koguarv.

```
> sample_n(sonad, 100) %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 x 1
      n
  <int>
1    15

> sample_n(sonad, 100) %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 x 1
      n
  <int>
1    10
```

prop.test

Uuringutes küllalt sageli ongi nii, et meil on võimalik mõõta mingi osa objekte ning saadud tulemuste põhjal tuleb teha võimalikult hea ehk siis arusaadavate usalduspiiridega järeldus kõigi sarnaste objektide peale. R-is käsklus ühe osakomponendi suhtarvupiiride leidmisks käsklus prop.test - jälgime, mida vastusest välja lugeda võib. Näitena olukord, kus Kungla rahva sõnu oli sajast valitud sõnast 15.

```
> prop.test(15, 100)

        1-sample proportions test with continuity correction

data:  15 out of 100, null probability 0.5
X-squared = 47.61, df = 1, p-value = 5.2e-12
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.0891491 0.2385308
sample estimates:
```

```
   p
0.15
```

Esialgu on saadud vastusest tähtsaim lause: "95% tõenäosusega võime väita, et nähtud andmete põhjal jääb Kungla rahva sõnade osakaal kõigis uuritud sõnadest 8,9% kuni 23,9% vahele.

Samuti võrreldakse andmeid nullhüpoteesiga (kahes tekstis on võrdselt sõnu) ning teatatakse, et selle kehtivuse tõenäosus on 5.2e-12, ehk siis 0,0000000000052 - mis on sama vähe tõenäoline, kui peidetud liivatera esimesel katsel leidmine tuhande kuupmeetri liiva seest.

Teisel katsel leitud kümme sõna sajast seab Kungla rahva sõnade eeldatavaks osakaaluks 0,05 kuni 0,18. Nagu näha, siis vahemik mõnevõrra kõigub, aga 8 kuni 18 protsenti on mõlema katse puhul ennustatava vahemiku sees.

```
> prop.test(10, 100)

        1-sample proportions test with continuity correction

data:  10 out of 100, null probability 0.5
X-squared = 62.41, df = 1, p-value = 2.789e-15
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.0516301 0.1803577
sample estimates:
  p
0.1
```

Kui katsel juhtus tulema kaheksa sõna sajast,

```
> prop.test(8, 100)
95 percent confidence interval:
 0.03767874 0.15614533
```

siis eeldatav vahemik läks nelja ja 16 protsendi vahele.

Usaldusintervall

Sõltuvalt uuringu tulemuste mõjust vajatakse eri täpsusega hinnanguid. Päris täpse tulemuse saame vaid siis, kui kõik sõnad kummaski tekstis üle loeme. Kui see aga mingil põhjusel jõukohane pole ja oleme valmis leppima vale hinnanguga ühel juhul sajast, siis võime sättida usaldusnivoo (conf.level) 99% peale ja vaadata, millist usaldusintervalli ehk -vahemikku meile pakutakse.

```
> prop.test(8, 100, conf.level=0.99)

99 percent confidence interval:
 0.03062124 0.18502380
```

Võrdlusena - eelnevalt 95% juures oli vahemik `0.03767874 0.15614533` - ehk siis mõnevõrra kitsam. Nii ka teiste mõõtmiste puhul - lihtsalt vahemik ise veidi teises kohas:

```
> prop.test(15, 100, conf.level=0.99)

99 percent confidence interval:
 0.07652508 0.26925703
```

Ka väiksem katsete arv laiendab intervalli - ning suurem vastupidi kahandab. Näiteks kui tuleb ühel korral kümnest sõna "Kungla rahvast", siis selle järgi võin 99% tõenäosusega väita vaid, et Kungla rahva sõnu on kõikide sõnade hulgas 0,3 kuni 55 protsenti.

```
> prop.test(1, 10, conf.level=0.99)

99 percent confidence interval:
 0.003298139 0.554819141
```

Kui sõnu on kümme sajast, siis piisab sellest 99% tõenäosusega väitmiseks, et sõnade osakaal on 4 kuni 21 protsenti

```
> prop.test(10, 100, conf.level=0.99)

99 percent confidence interval:
 0.04284027 0.20989670
```

Tõehetk

Seni mängisime andmetega pimesikku. Siin näites aga meil üldkogum olemas ning võimalik ka tegelik vastus kätte saada. Kungla rahva sõnade suhtarv on:

```
> sonad %>% filter(lugu=="kungla") %>% count() / nrow(sonad)
          n
1 0.1116071
```

Mõlema loo sõnade üldarv:

```
> sonad %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 2 x 2
  lugu      kogus
  <chr>     <int>
1 kungla       75
2 lambipirn   597
```

Kui teha katseid suuremate arvudega, siis tulevad need juba leitud suhte lähedale. Kui `sample_n` käsu puhul lisada parameeter `replace=TRUE`, siis lubatakse kord võetud ridu ka hiljem kuvada.

```
> sample_n(sonad, 1000, replace=TRUE) %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 x 1
      n
  <int>
1   123

> sample_n(sonad, 10000, replace=TRUE) %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 x 1
      n
  <int>
1  1129
```

Kui katsete tulemusena saadud, et 1129 rida kümnest tuhandest on mingit tüüpi, siis sealtkaudu läheb usaldusvahemik juba võrreldes eelmistega päris kitsaks kokku ning nagu nüüd meile teada olevate andmetega võrrelda saab, siis näitab ka õigesti.

```
> prop.test(1129, 10000)
95 percent confidence interval:
 0.1067966 0.1193031
```

Võrdlus olemasoleva suhtega

Test võimaldab katsete põhjal leitud suhet võrrelda varem arvutatud suhtega ning näidata, kui tõenäoline on, et andmed võiksid olla samast üldkogumist ehk "ühisest potist". Leiame Lambipirni jutus viietäheliste ja pikemate sõnade osakaalu:

```
 sonad %>% filter(lugu=="lambipirn" & sonapikkus>=5) %>% count() /
    sonad %>% filter(lugu=="lambipirn") %>% count()
          n
1 0.6549414
```

Leiame Kungla rahva loos vähemalt viietähelised sõnad ning sõnade üldarvu.

```
> sonad %>% filter(lugu=="kungla" & sonapikkus>=5) %>% count()
# A tibble: 1 x 1
      n
  <int>
1    37
> sonad %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 x 1
      n
  <int>
1    75
```

Nüüd võrdleme andmeid omavahel.

```
> prop.test(37, 75, p=0.65)

        1-sample proportions test with continuity correction
```

```
data:  37 out of 75, null probability 0.65
X-squared = 7.4176, df = 1, p-value = 0.006459
alternative hypothesis: true p is not equal to 0.65
95 percent confidence interval:
 0.3769863 0.6103674
sample estimates:
        p
0.4933333
```

Püüame tulemuse inimkeeli kirja panna. Võrdleme Kungla rahva pikkade sõnade suhet 37 75st (mille puhul me ei eelda, et kõik sõnad on teada) Lambipirni jutu eelnevalt teadaoleva 0.65-ga. Üleval näidatud p-väärtus 0.006459 tähendab, et tõenäosus, et sõnade pikkus ei sõltu loost on vaid 0,65%. 99,35% tõenäosusega on järelikult seos olemas. Kui meil on teada 37 sõna kohta 75st, et nood on vähemasti viie tähe pikkused, siis selle põhjal saab 95% tõenäosusega väita, et uuritava teksti vähemalt 5-täheliste sõnade osa on 38% kuni 61%.

Harjutus

- Loe kokku, mitu kuni kolmetähelist sõna on Eesti hümni esimeses lauses. Näita prop.test-i abil, mida võiks selle põhjal järeldada kogu hümni sõnade pikkuse kohta
- Muuda usaldusintervalli, näita kuidas sellega koos muutub usaldusvahemik.

Mu isamaa, mu õnn ja rõõm, kui kaunis oled sa

10 sõna
6 kuni kolmetähelist

```
 prop.test(6, 10)

        1-sample proportions test with continuity correction

data:  6 out of 10, null probability 0.5
X-squared = 0.1, df = 1, p-value = 0.7518
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.2736697 0.8630694
sample estimates:
  p
0.6


> prop.test(6, 10, conf.level =0.99)

        1-sample proportions test with continuity correction

data:  6 out of 10, null probability 0.5
X-squared = 0.1, df = 1, p-value = 0.7518
alternative hypothesis: true p is not equal to 0.5
99 percent confidence interval:
 0.2152334 0.8972854
sample estimates:
  p
0.6
```

```
> prop.test(6, 10, conf.level=0.75)

        1-sample proportions test with continuity correction

data:  6 out of 10, null probability 0.5
X-squared = 0.1, df = 1, p-value = 0.7518
alternative hypothesis: true p is not equal to 0.5
75 percent confidence interval:
 0.3739852 0.7964647
sample estimates:
  p
0.6
```

Vahemike graafiline kuvamine

Eri katsetel saadud vahemikke võib olla vahel kasulik illustreerimiseks kuvada joonisel. Selleks loome kõigepealt mõned andmed ja siis kuvame. Kõigepealt meeldetuletus katsest, kus väljundiks oli kümme Kungla rahva sõna saja sõna peale kokku. Käsk prop.test väljastas selle peale, et sõnade esinemissagedus üldkogumis võiks olla 5% kuni 18%

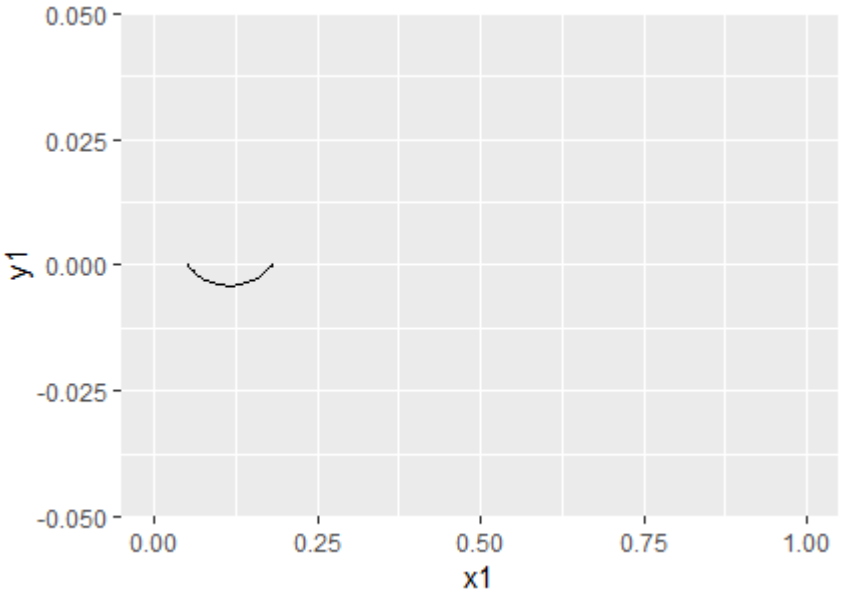
```
> prop.test(10, 100)

        1-sample proportions test with continuity correction

data:  10 out of 100, null probability 0.5
X-squared = 62.41, df = 1, p-value = 2.789e-15
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.0516301 0.1803577
sample estimates:
  p
0.1
```

Andmete mugavamaks sisestamiseks joonisekäsklusesse koostame neist koordinaatide üherealise tabeli

```
koordinaadid=tibble(x1=0.051, y1=0, x2=0.18, y2=0)
```

Käsuga geom_curve tõmbame nende kahe punkti vahele kõverjoone

```
ggplot() + xlim(0, 1) +
   geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
              data=koordinaadid)
```



Sama katse kohta, kus saadi 15 Kungla rahva sõna sajast

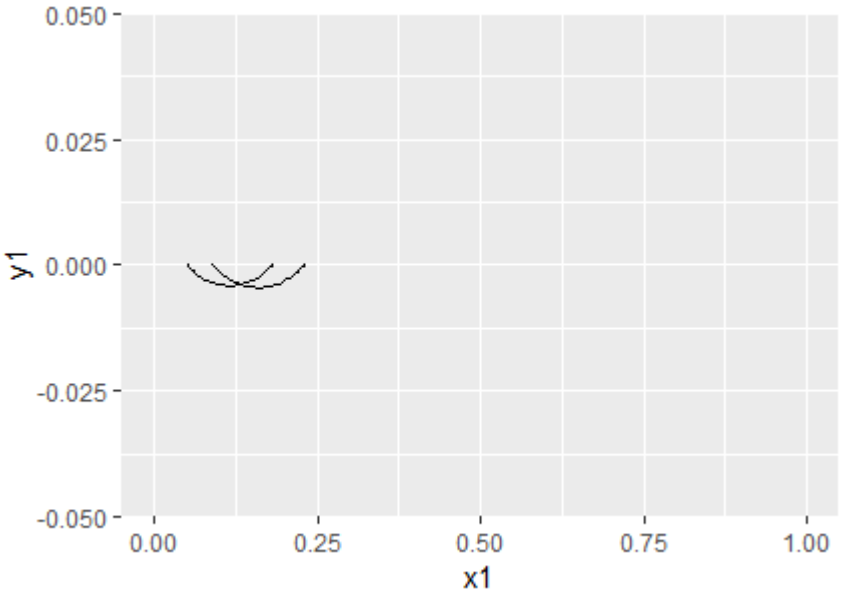
```
> prop.test(15, 100)

95 percent confidence interval:
 0.0891491 0.2385308
```

Tabelisse paneme mõlemad väärtused koos. Kõigepealt esimene ning siis `add_row` käsu abil teine juurde

```
koordinaadid=tibble(x1=0.051, y1=0, x2=0.18, y2=0)
koordinaadid <- koordinaadid %>%  add_row(x1=0.089, y1=0, x2=0.23, y2=0)
koordinaadid
# A tibble: 2 x 4
     x1    y1    x2    y2
  <dbl> <dbl> <dbl> <dbl>
1 0.051     0  0.18     0
2 0.089     0  0.23     0
```

`ggplot` suudab mõlema rea põhjal kaared joonistada, `xlim(0, 1)` näitab x-telje ulatust joonisel.

```
> ggplot() + xlim(0, 1) +
+   geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
+              data=koordinaadid)
```



Automatiseeritud joonis

Üksikud väärtused kannatab ekraanilt lugeda ning sinna sobivatesse kohtadesse tagasi toksida. R võimaldab testide väljundid ka muutujate kaudu kätte saada ning nii on võimalik neid otse edasi toimetada ja sealtkaudu lasta arvutil hulga arvutusi järgemööda ette võtta. Testi pakutava vahemiku alumise ja ülemise piiri saab kätte `conf.int`-muutuja kaudu

```
> prop.test(10, 100)$conf.int[1]
[1] 0.0516301
> prop.test(10, 100)$conf.int[2]
[1] 0.1803577
```

Lihtsalt ridade arvu küsides annab `count`-käsklus vastuseks tabeli, kus on üks tulp nimega `n` ning selle sees on üks rida saadud väärtusega.

```
> sonad %>% sample_n(100) %>% filter(lugu=="kungla") %>% count()
# A tibble: 1 × 1
      n
  <int>
1     9
```

Valemisse on aga ainult seda arvu vaja, mitte tervet tabelit. Väärtuse küsimiseks tuleb käsuahelale lisada veel üks etapp - punkt tähistab jooksvat andmestikku ning sellele järgnev dollar ja täht tulpa, millest soovitakse väärtus saada. Nii tulebki soovitud arv otse esile.

```
> sonad %>% sample_n(100) %>% filter(lugu=="kungla") %>% count() %>% .$n
[1] 6
```

Juhuslike ridade valimise tõttu `sample_n` käsus on uuel käivitamisel tulemus midagi muud

```
> sonad %>% sample_n(100) %>% filter(lugu=="kungla") %>% count() %>% .$n
[1] 14
```

Saadud arvu saab prop.test-käsus juba sobivale kohale paigutada.

```
> prop.test(sonad %>% sample_n(100) %>% filter(lugu=="kungla") %>% count() %>% .$n, 100)

        1-sample proportions test with continuity correction

data:  sonad %>% sample_n(100) %>% filter(lugu == "kungla") %>% count() %>%  out of 100,
null probability 0.5    .$n out of 100, null probability 0.5
X-squared = 68.89, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.03767874 0.15614533
sample estimates:
   p
0.08
```

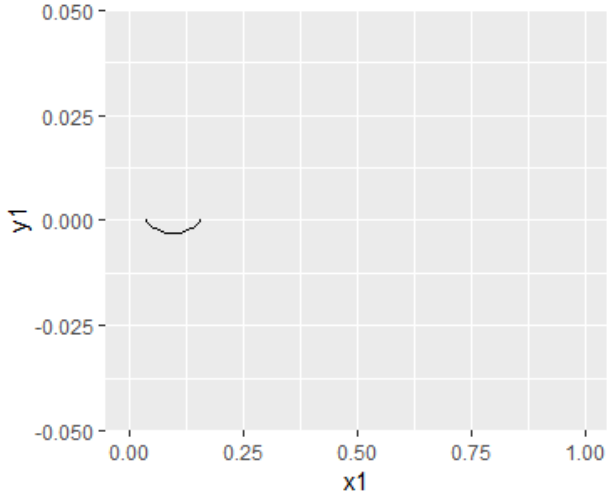
Või siis muutujasse lugeda ning sealt juba vastuste tabeli kaudu joonisele paigutada.

```
testivastus <- prop.test(sonad %>% sample_n(100) %>%
                           filter(lugu=="kungla") %>% count() %>% .$n,
                           100)
koordinaadid=tibble(x1=testivastus$conf.int[1], y1=0, x2=testivastus$conf.int[2], y2=0)
koordinaadid
ggplot() + xlim(0, 1) +
  geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
             data=koordinaadid)
```

Vastusena näha kinni püütud koordinaadid

```
# A tibble: 1 x 4
      x1    y1    x2    y2
   <dbl> <dbl> <dbl> <dbl>
1 0.0377     0 0.156     0
```

ning neile vastav joonis



Kordused

Soovitud arvuloetelu saab, kui arvude ja kooloni abil ette anda vahemik

```
> 1:5
[1] 1 2 3 4 5
```

Nõnda palju kordi mõnd tegevust korrata kannatab for-tsükliga. Praegusel juhul kuvatakse vastavate arvude ruudud

```
for(x in 1:5){
  print(x*x)
}
```

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

Ühed ees, sest igal korral tuleb uus eraldi üheelemendiline vastus.

R-keeles on kordustega toimetamiseks lisaks tsüklitele olemas apply-perekonna funktsioonid, suhteliselt lihtsasti kasutatav neist `sapply`.

```
sapply(1:5, function(x){
  sonad %>% sample_n(100) %>%
    filter(lugu=="kungla") %>% count() %>% .$n
})
```

```
[1] 10 10  7 17 17
```

Uuel käivitamisel salvestame arvud eraldi muutujasse `kunglakogused` ning vaatame selle väärtust

```
kunglakogused <- sapply(1:5, function(x){
  sonad %>% sample_n(100) %>%
    filter(lugu=="kungla") %>% count() %>% .$n
})

kunglakogused
```

```
[1] 12 13  8 12  7
```

Nüüdse kordusega teeme iga leitud koguse peale `prop.test`-i ja testivastusteks x1 ja x2 kohale paigutame leitud usaldusintervalli alam- ja ülempiiri

```
testivastused=sapply(kunglakogused, function(kogus){
  pt=prop.test(kogus, 100)
  c(x1=pt$conf.int[1], y1=0, x2=pt$conf.int[2], y2=0)
```

```
})
testivastused
> testivastused
         [,1]       [,2]       [,3]       [,4]       [,5]
x1 0.06625153 0.07376794 0.03767874 0.06625153 0.03101985
y1 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
x2 0.20397718 0.21560134 0.15614533 0.20397718 0.14376573
y2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
```

Joonise mugavamaks koostamiseks vahetame read ja veerud

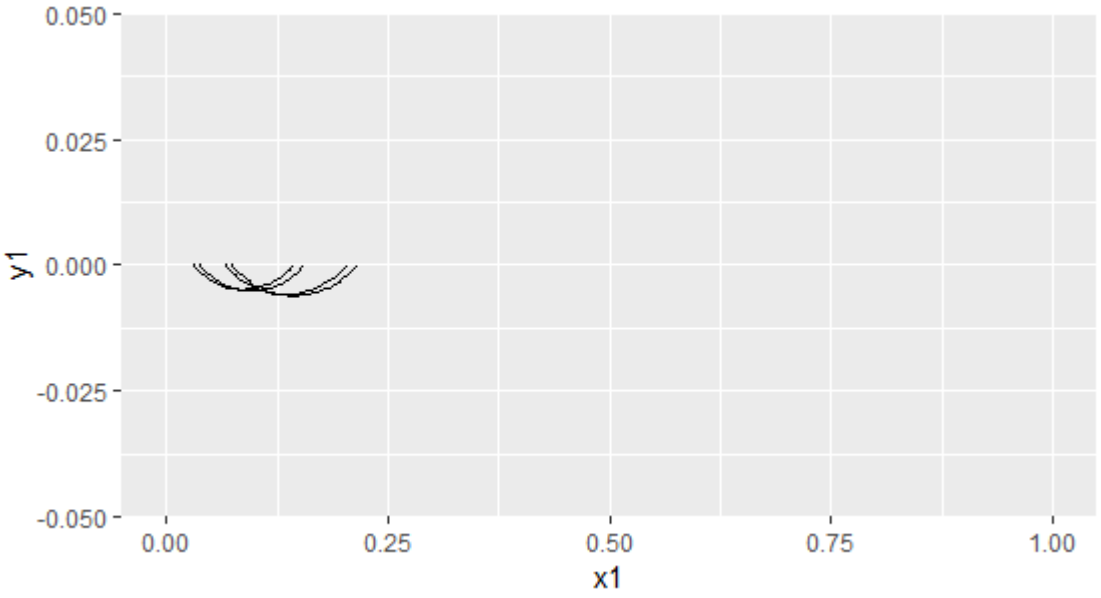
```
> t(testivastused)
             x1 y1        x2 y2
[1,] 0.06625153  0 0.2039772  0
[2,] 0.07376794  0 0.2156013  0
[3,] 0.03767874  0 0.1561453  0
[4,] 0.06625153  0 0.2039772  0
[5,] 0.03101985  0 0.1437657  0
```

ning muudame uuemale tibble-kujule

```
> as_tibble(t(testivastused))
# A tibble: 5 x 4
      x1    y1    x2    y2
   <dbl> <dbl> <dbl> <dbl>
1 0.0663     0 0.204     0
2 0.0738     0 0.216     0
3 0.0377     0 0.156     0
4 0.0663     0 0.204     0
5 0.0310     0 0.144     0
```

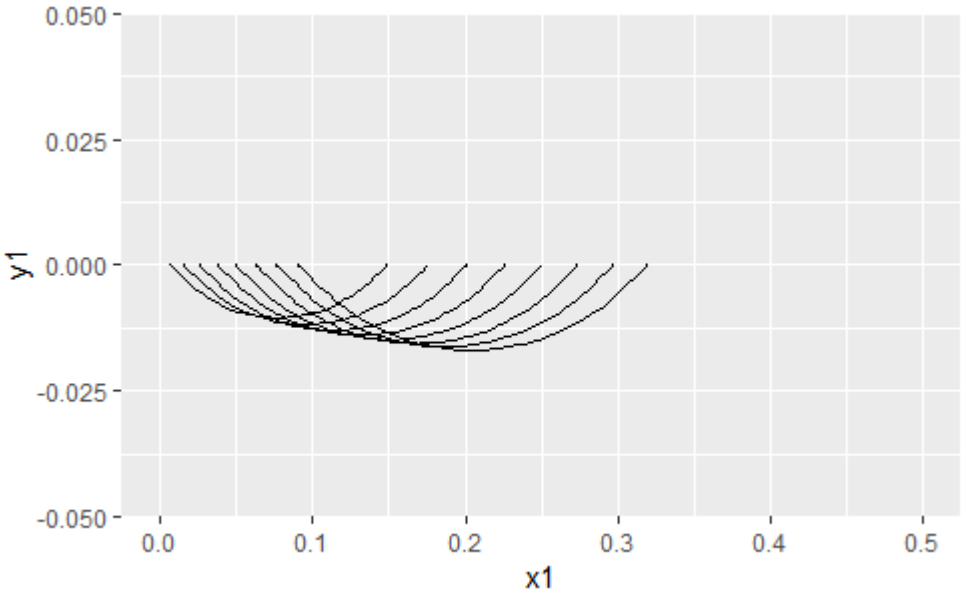
Edasi saame sealt ggplot-i ja selle sees kõverjoone loova geom_curve abil kaared joonisele kätte

```
ggplot() + xlim(0, 1) +
  geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
             data=as_tibble(t(testivastused)))
```



Sama arvutus üldisemal kujul, kus yldarv tähendab, et mitu juhuslikku sõna kogumist võetakse ning katsete arv, et mitu korda vastavat katset korratakse. Taas loetakse iga katse korral, et mitu sõna oli Kungla rahvast ning joonistatakse välja selle põhjal kättesaadavad üldistuspiirid

```
yldarv <- 50
katsetearv <- 20

sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
kunglakogused <- sapply(1:katsetearv, function(x){
  sonad %>% sample_n(yldarv) %>%
    filter(lugu=="kungla") %>% count() %>% .$n
})


testivastused <- sapply(kunglakogused, function(kogus){
  pt=prop.test(kogus, yldarv)
  c(x1=pt$conf.int[1], y1=0, x2=pt$conf.int[2], y2=0)
})

ggplot() + xlim(0, 0.5) +
  geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
             data=as_tibble(t(testivastused)))



kunglakogused
 [1] 6 2 4 8 5 6 3 7 6 8 7 5 4 4 8 5 2 4 9 2
```



Harjutus

- Katsetage prop.test käsku. Öelge inimkeeles välja, mida võib järeldada Lambipirni sõnade suhtarvu kohta, kui neid tuli 9 tk kümnest. Muutke usaldusnivoo 99% peale ja vaadake siis usaldusvahemikku.
- Kui kümnest sõnast tuli 9 lambipirni jutust - kui suur on tõenäosus, et lambipirni sõnu võiks olla 75%?

- Kohandage kaarte joonistamise näidet, muutke katsete arvu ja valimi suurust, jälgige tulemuse muutusi. Proovige mitmel korral joonisele kuvada 100 kaart - lugege, mitmel korral jäi Kungla rahva sõnade tegelik osakaal (11%) usaldusvahemikust välja
- Filtreerige välja Lambipirni sõnu, näidake, millised vahemikud siis tulevad

```
> prop.test(9, 10)

        1-sample proportions test with continuity correction

data:  9 out of 10, null probability 0.5
X-squared = 4.9, df = 1, p-value = 0.02686
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5411540 0.9947577
sample estimates:
  p
0.9
```

Katse tulemusena võin järeldada, et 95% tõenäosusega on sõnade üldkogumis Lambipirni loo sõnu 54%-99%

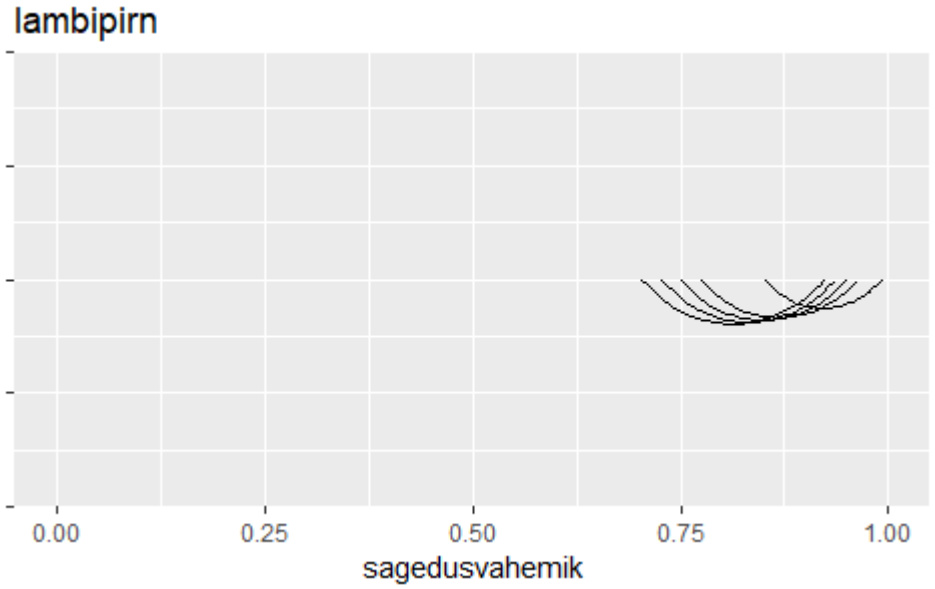
```
> prop.test(9, 10, conf.level=0.99)

        1-sample proportions test with continuity correction

data:  9 out of 10, null probability 0.5
X-squared = 4.9, df = 1, p-value = 0.02686
alternative hypothesis: true p is not equal to 0.5
99 percent confidence interval:
 0.4451809 0.9967019
sample estimates:
  p
0.9
```

99% tõenäosusega saan väita, et Lambipirni loo sõnu on üldkogumis 44,5%-99,7%

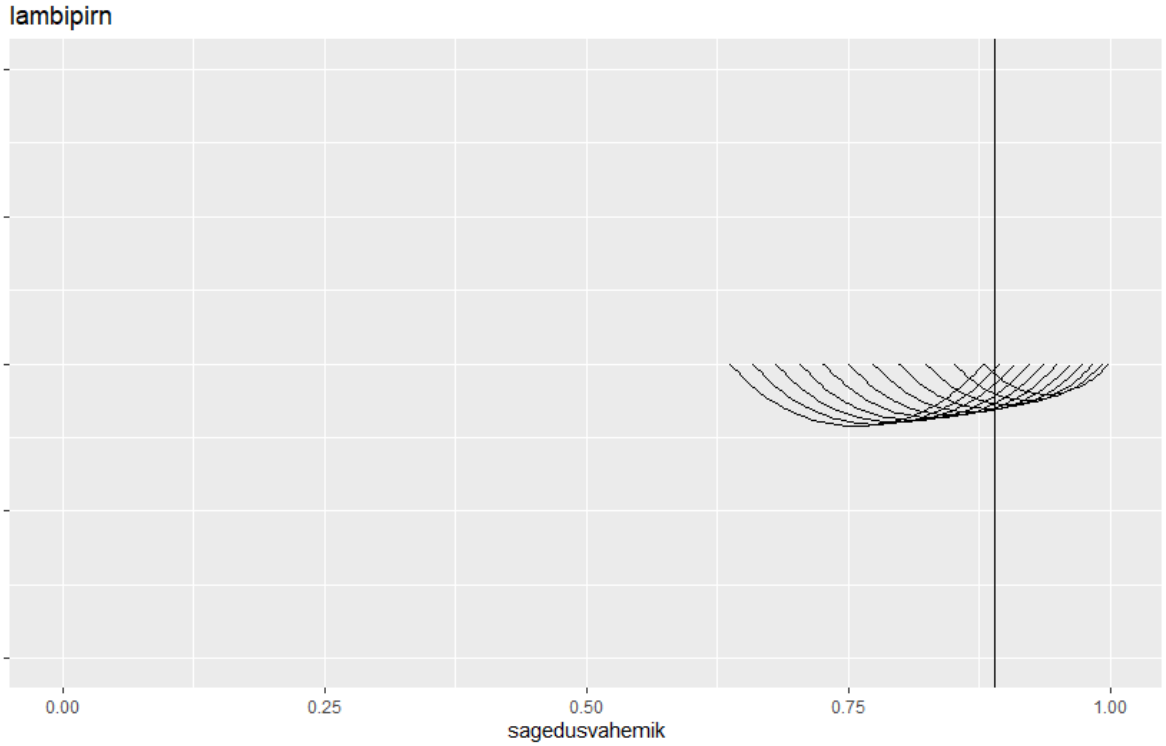
```
> prop.test(9, 10, p=0.75)

        1-sample proportions test with continuity correction

data:  9 out of 10, null probability 0.75
X-squared = 0.53333, df = 1, p-value = 0.4652
alternative hypothesis: true p is not equal to 0.75
95 percent confidence interval:
 0.5411540 0.9947577
sample estimates:
  p
0.9
```

Tõenäosus, et andmed pärinevad üldkogumist, kus Lambipirni sõnu on 75% on 47% (ehk ei saa sinna kogumisse kuulumist välistada ega kinnitada)

```
yldarv <- 50
katsetearv <- 10
loonimi <- "lambipirn"

lookogused <- sapply(1:katsetearv, function(x){
  sonad %>% sample_n(yldarv) %>%
    filter(lugu==loonimi) %>% count() %>% .$n
})


testivastused <- sapply(lookogused, function(kogus){
  pt=prop.test(kogus, yldarv)
  c(x1=pt$conf.int[1], y1=0, x2=pt$conf.int[2], y2=0)
})

ggplot() + xlim(0, 1.0) +
  geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
        data=as_tibble(t(testivastused)))+
        ggtitle(loonimi) +
         xlab("sagedusvahemik") + ylab("") +
         theme(axis.text.y=element_blank())

lookogused

 [1] 45 45 48 45 42 43 48 43 44 44
```




```
ggplot() + xlim(0, 1.0) + ylim(-1, 1)+
  geom_curve(aes(x=x1, y=y1, xend=x2, yend=y2),
             data=as_tibble(t(testivastused)))+
    geom_vline(xintercept=0.89)+
  ggtitle(loonimi) +
  xlab("sagedusvahemik") + ylab("") +
  theme(axis.text.y=element_blank())
```



2x2 tabel

Märkimisväärne hulk uuringuid jõuab andmete esitamise juures 2x2 tabelini. Siin näites kaks lugu ning loendamine, et kui palju on vähemalt viietähelisi sõnu ning kui palju sellest lühemaid.

```
sonapikkused <- sonad %>% group_by(lugu) %>%
  summarise(
   lyhikesi=sum(sonapikkus<5),
   pikki=sum(sonapikkus>=5)
  ) %>% ungroup()

> sonapikkused
# A tibble: 2 x 3
  lugu      lyhikesi pikki
  <chr>        <int> <int>
1 kungla          38    37
2 lambipirn      206   391
```

Testi saab teha vaid arvulistele tunnustele, nii eemaldame tabelist loo nimed

```
> sonapikkused %>% select(-lugu)
# A tibble: 2 x 2
  lyhikesi pikki
     <int> <int>
1       38    37
2      206   391
```

ja teisendame prop.test käsu jaoks sobilikule maatriksi kujule

```
> sonapikkused %>% select(-lugu) %>% as.matrix()
     lyhikesi pikki
[1,]       38    37
[2,]      206   391
```

Test ise:

```
> prop.test(sonapikkused %>% select(-lugu) %>% as.matrix())

        2-sample test for equality of proportions with continuity
        correction

data:  sonapikkused %>% select(-lugu) %>% as.matrix()
X-squared = 6.8422, df = 1, p-value = 0.008903
alternative hypothesis: two.sided
95 percent confidence interval:
 0.03470217 0.28851392
sample estimates:
   prop 1    prop 2
0.5066667 0.3450586
```

Seletus:

- Tõenäosus, et lugudes kasutatakse sarnase pikkusega sõnu on 0.008903 (alla ühe protsendi), 99% tõenäosus, et sõnade pikkused lugudes on erinevad

- Olemasolevate andmete puhul saab 95% tõenäosusega väita, et esimeses loos (Kungla rahvas) on alla 5 tähe pikkusi sõnu 3,4 kuni 28,9 protsendipunkti võrra rohkem kui teises loos (Lambipirn).

Andmestike sarnasuse katsetus. Näide, kus suhted on märgatavalt erinevad - 20/10 vs 80/290

```
> matrix(nrow=2, ncol=2, c(20, 10, 80, 290))
     [,1] [,2]
[1,]   20   80
[2,]   10  290


> prop.test(matrix(nrow=2, ncol=2, c(20, 10, 80, 290)))

       2-sample test for equality of proportions with continuity correction

data:  matrix(nrow = 2, ncol = 2, c(20, 10, 80, 290))
X-squared = 27.676, df = 1, p-value = 1.435e-07
alternative hypothesis: two.sided
95 percent confidence interval:
 0.07901275 0.25432059
sample estimates:
    prop 1     prop 2
0.20000000 0.03333333
```

Tõenäosus, et andmed võiksid samast üldkogumist olla on 0,00000014 ehk suhteliselt olematu

Teine andmestik, kus suhted sarnased (20/10) vs (80/40)

```
> prop.test(matrix(nrow=2, ncol=2, c(20, 10, 80, 40)))

       2-sample test for equality of proportions without continuity correction

data:  matrix(nrow = 2, ncol = 2, c(20, 10, 80, 40))
X-squared = 0, df = 1, p-value = 1
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.1357903  0.1357903
sample estimates:
prop 1 prop 2
   0.2    0.2
```

Nullhüpoteesi tõenäosus 100%, ehk miski ei lükka ümber, et andmed võiksid samast üldkogumist olla - samas see otseselt ka ei kinnita seda. 95% tõenäosusega võib sisaldus võrreldud valimite üldkogumites erineda +/- 13%

Harjutus

- Võta Lambipirni loost kahel juhul sada juhuslikku sõna ja näita, mitme sõna pikkused olid alla 5 tähe. Võrdle tulemusi prop.testi abil. Kui suure tõenäosusega loetakse andmestikud sarnaseks?

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
sonad %>% filter(lugu=="lambipirn") %>% sample_n(100) %>%
    mutate(pikk=sonapikkus<5) %>% group_by(pikk) %>% summarise(kogus=n())

  pikk  kogus
  <lgl> <int>
1 FALSE    62
2 TRUE     38


  pikk  kogus
  <lgl> <int>
1 FALSE    70
2 TRUE     30

> prop.test(matrix(nrow=2, ncol=2, c(62, 38, 70, 30)))

        2-sample test for equality of proportions with continuity correction

data:  matrix(nrow = 2, ncol = 2, c(62, 38, 70, 30))
X-squared = 1.0918, df = 1, p-value = 0.2961
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.24578818  0.06753506
sample estimates:
   prop 1    prop 2
0.4696970 0.5588235
```

Nullhüpoteesi ehk võrdse osakaalu tõenäosus on 29,6%. Ehkki andmed on üsna erinevad, ei anna see veel põhjust väita, et need tuleksid eri allikatest. 95% tõenäosusega eri allikatest tuleku väitmiseks peaks p-väärtus olema alla 5% ehk alla 0,05

```
> sonad %>% filter(lugu=="lambipirn") %>% sample_n(100) %>%
+     mutate(pikk=sonapikkus>=5) %>% group_by(pikk) %>% summarise(kogus=n())
# A tibble: 2 x 2
  pikk  kogus
  <lgl> <int>
1 FALSE    33
2 TRUE     67
>
> sonad %>% filter(lugu=="lambipirn") %>% sample_n(100) %>%
+     mutate(pikk=sonapikkus>=5) %>% group_by(pikk) %>% summarise(kogus=n())
# A tibble: 2 x 2
  pikk  kogus
  <lgl> <int>
1 FALSE    36
2 TRUE     64

> prop.test(matrix(nrow=2, ncol=2, c(33, 67, 36, 64)))
```

```
        2-sample test for equality of proportions with continuity correction

data:  matrix(nrow = 2, ncol = 2, c(33, 67, 36, 64))
X-squared = 0.088505, df = 1, p-value = 0.7661
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.1899208  0.1235418
sample estimates:
   prop 1    prop 2
0.4782609 0.5114504
```

Rohkem kui kaks mõõtmist

Siin on kolmel korral mõõdetud, mitu otsitavat (nt. Kungla rahva sõna) leiti saja objekti (sõna) hulgast. Leiti vastavalt katsele 8, 10 ja 15 korda. Käsu prop.test väljund näitab, et tõenäosus, et mõõdeti sisaldust samasugusest valimist on 26,5% (aga praegu oligi sama andmestik). Alles siis, kui p läheks alla 0.05 või 0.01 või mõne muu olulisusnivooks võetud suhte, võiksime hakata väitma, et sagedused rühmiti erinevad.

```
> prop.test(c(8, 10, 15), c(100, 100, 100))

        3-sample test for equality of proportions without continuity
        correction
data:  c(8, 10, 15) out of c(100, 100, 100)
X-squared = 2.6558, df = 2, p-value = 0.265
alternative hypothesis: two.sided
sample estimates:
prop 1 prop 2 prop 3
  0.08   0.10   0.15
```

Järgmises näites võtame (Kungla rahva) sõnad kümne kaupa rühmadesse ja koostame nõnda laulu esimesest viiekümnest sõnast viis rühma, kus arvutame sõnapikkuste ning sulghäälikute arvu summad.

```
kogused <- sonad %>% mutate(ryhm=floor(row_number()/10)) %>%
    filter(ryhm<5) %>% group_by(ryhm) %>%
    summarise(pikkus=sum(sonapikkus), sulgh=sum(sulghaalikuid))

kogused
```

```
> kogused
# A tibble: 5 x 3
   ryhm pikkus sulgh
  <dbl>  <int> <int>
1     0     43     8
2     1     52     7
3     2     51     9
4     3     42     3
5     4     51     8
```

Nii saab kogused eraldi välja küsida, nagu näha, siis pikkuste summad on rühmas erinevad

```
> kogused$sulgh
[1] 8 7 9 3 8
> kogused$pikkus
[1] 43 52 51 42 51
```

Testi tulemus ütleb, et tõenäosus, et valimid on samast üldkogumist on 0,5746, ehk siis pole põhjust kahtlustama hakata, et laulu tagumistes lõikudes sulghäälikute osakaal oluliselt erineks esimeste lõikude omadest.

```
> prop.test(kogused$sulgh, kogused$pikkus)

        5-sample test for equality of proportions without continuity
        correction

data:  kogused$sulgh out of kogused$pikkus
X-squared = 2.9007, df = 4, p-value = 0.5746
alternative hypothesis: two.sided
sample estimates:
    prop 1     prop 2     prop 3     prop 4     prop 5
0.18604651 0.13461538 0.17647059 0.07142857 0.15686275
```

Võtmesõnade leidmine

Teksti lugemine veebist. Edasi väiketähtedeks, kõik arvudest ja tähtedest erinevad sümbolid tühikuteks. Kantsulgudes 1 `gsub` käskluse taga on vajalik, kuna käsk võib korraga tegutseda ka mitme tekstiga - praegu küsime tagasi selle esimese ja ainukese. Tekst tühikute koha pealt sõnadeks - jälle samal põhjusel kantsulud taga. Käsk str_length näitab sõnade pikkusi. Jätame alles vaid sõnad, mille pikkus on suurem kui 0. Muudame loetelu tibbleks, rühmitame sõnade järgi, loeme iga sõna kohta kokku et mitu sõna on ning järjestame koguse järgi kahanevalt.

```
tekst=read_file("http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/Johnson_OL.txt")
tekst=str_to_lower(tekst)
tekst=gsub("[^a-zõäöü0-9]", " ", tekst)[[1]]
tsonad=str_split(tekst, " ")[[1]]
head(tsonad)
str_length(tsonad)
tsonad=tsonad[str_length(tsonad)>0]
head(tsonad)
sagedused1=tibble(sona=tsonad) %>% group_by(sona) %>% summarise(kogus=n()) %>%
   arrange(desc(kogus))
head(sagedused1)
 sona    kogus
  <chr>   <int>
1 et        124
2 on        101
3 johnson    92
4 ja         72
5 kui        63
6 ei         59
```

Sama teise failiga

```
tekst=read_file("http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/Johnson_PM.txt")
tekst=str_to_lower(tekst)
tekst=gsub("[^a-zõäöü0-9]", " ", tekst)[[1]]
tsonad=str_split(tekst, " ")[[1]]
head(tsonad)
str_length(tsonad)
tsonad=tsonad[str_length(tsonad)>0]
head(tsonad)
sagedused2=tibble(sona=tsonad) %>% group_by(sona) %>% summarise(kogus=n()) %>%
  arrange(desc(kogus))
head(sagedused2)
```

```
  sona    kogus
  <chr>   <int>
1 on        108
2 ja        100
3 et         85
4 johnson    83
5 brexiti    70
6 boris      63
```

Tabelid kõrvuti

head(sagedused1, 20) %>% add_column(PM=head(sagedused2, 20))

```
# A tibble: 20 x 3
   sona          kogus PM$sona          $kogus
   <chr>         <int> <chr>             <int>
 1 et              124 on                  108
 2 on              101 ja                  100
 3 johnson          92 et                   85
 4 ja               72 johnson              83
 5 kui              63 brexiti              70
 6 ei               59 boris                63
 7 ta               52 ei                   60
 8 boris            47 briti                56
 9 euroopa          44 peaminister          54
10 ning             43 kui                  44
11 suurbritannia    41 johnsoni             41
12 brexiti          38 euroopa              37
13 ka               38 ta                   33
14 peaminister      32 mis                  32
15 leppeta          27 ka                   31
16 seda             27 leppeta              29
17 parlamendi       26 parlamendi           29
18 oma              25 oma                  28
19 31               24 vastu                28
20 johnsoni         24 ühendkuningriigi     28
```

Tabelid sõna kaupa kõrvuti, tühjad tulbad arvulisteks nullideks

```
sagedused=sagedused1 %>% full_join(sagedused2, by="sona") %>%
  replace_na(list("kogus.x"=0, "kogus.y"=0))

>  head(sagedused)
```

```
# A tibble: 6 x 3
  sona    kogus.x kogus.y
  <chr>     <dbl>   <dbl>
1 et          124      85
2 on          101     108
3 johnson      92      83
4 ja           72     100
5 kui          63      44
6 ei           59      60
```

Lõpuotsas näha nulle

```
>  tail(sagedused)
# A tibble: 6 x 3
  sona          kogus.x kogus.y
  <chr>           <dbl>   <dbl>
1 ülevaatlikuma       0       1
2 ülikooli            0       1
3 ülimat              0       1
4 üritanud            0       1
5 yellowhammer        0       1
6 yougov              0       1
```

Sõnade arvud tekstides

```
> nrow(sagedused1)
[1] 2154
> nrow(sagedused2)
[1] 2479
```

Sõna johnson kasutuste erinevus:

```
> prop.test(c(92, 83), c(2154, 2479))

data:  c(92, 83) out of c(2154, 2479)
X-squared = 2.4535, df = 1, p-value = 0.1173
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.002297231  0.020757216
sample estimates:
    prop 1     prop 2
0.04271123 0.03348124
```

Käsu abiinfo küsimine

```
> ?prop.test
```

Sealt sai teada, et tunnusega p.value on võimalik vastav väärtus otse kätte saada

```
> prop.test(c(92, 83), c(2154, 2479))$p.value
[1] 0.1172599
```

prop.test tulemuste arvutamine, esialgu esimese viie rea kohta sageduste tabelis

```
sapply(1:5, function(nr){
  prop.test(c(sagedused$kogus.x[nr], sagedused$kogus.y[nr]), c(2154, 2479))$p.value})

[1] 0.000186301 0.636439861 0.117259861 0.244688701 0.012388196
```

ja terve tabeli kohta

```
pvaartused=sapply(1:nrow(sagedused), function(nr){
  prop.test(c(sagedused$kogus.x[nr], sagedused$kogus.y[nr]), c(2154, 2479))$p.value})
sagedused$p=pvaartused
head(sagedused)

sona    kogus.x kogus.y        p
  <chr>     <dbl>   <dbl>    <dbl>
1 et          124      85 0.000186
2 on          101     108 0.636
3 johnson      92      83 0.117
4 ja           72     100 0.245
5 kui          63      44 0.0124
6 ei           59      60 0.555
```

Järjestus p-väärtuse järgi - suurema erinevusega sõnad eespool

```
sagedused %>% arrange(p)
# A tibble: 51 x 4
   sona             kogus.x kogus.y         p
   <chr>              <dbl>   <dbl>     <dbl>
 1 briti                 14      56 0.0000132
 2 eli                   18       0 0.0000154
 3 suurbritannia         41      16 0.000183
 4 et                   124      85 0.000186
 5 ent                   13       0 0.000324
 6 ühendkuningriigi       5      28 0.000566
 7 erakorralised         12       0 0.000601
 8 ütles                  4      23 0.00183
 9 teemal                10       0 0.00208
10 ma                     0      12 0.00325
```

Sõnad, mille erinevuse p-väärtus on alla 0,05 - ehk siis mille kasutus tekstides on vähemasti 95% tõenäosusega erinev

```
> sagedused %>% arrange(p) %>% filter(p<0.05) %>% .$sona
 [1] "briti"            "eli"              "suurbritannia"    "et"
 [5] "ent"              "ühendkuningriigi" "erakorralised"    "ütles"
 [9] "teemal"           "ma"               "suurbritannias"   "downing"
[13] "juhul"            "järgi"            "ning"             "ta"
[17] "kui"              "jo"               "nende"            "19"
[21] "kriitikud"        "kuningannale"     "juba"             "valimised"
[25] "vastu"            "brexiti"          "lausus"           "seaduseelnõu"
[29] "streeti"          "erakorraliste"    "mitmed"           "nimetas"
[33] "toimuvad"         "000"              "hunt"             "eelnõu"
[37] "me"               "el"               "aastal"           "i"
[41] "enda"             "ise"              "küll"             "teisipäeval"
[45] "seda"             "ratas"            "meie"             "kokku"
[49] "ajapikendust"     "corbyn"           "seotud"
```

Harjutus

- Tehke näide läbi
- Leidke sarnasuse tõenäosus (p-väärtus) sõna "et" puhul. Ühes artiklis esineb 124 korda 2154 sõna hulgas, teises 85 korda 2479 sõna hulgas.
- Arvutage tabelisse igale sõnale juurde osakaal tekstis
- Kuvage välja sõnad, mille erinevuste p-väärtus on alla 0,05 ning mille osakaal on suurem Õhtulehe artiklites
- Tehke sama Postimehe artiklitega

```
sagedused %>% filter(p<0.02) %>%
   mutate(osa_oleht=kogus.x/nrow(sagedused1), osa_pm=kogus.y/nrow(sagedused2)) %>%
   mutate(ol_rohkem=osa_oleht>osa_pm) %>% group_by(ol_rohkem) %>%
   summarise(loetelu=paste(sona, collapse = ","))


# A tibble: 2 x 2
  ol_rohkem loetelu
  <lgl>     <chr>
1 FALSE     briti,vastu,ühendkuningriigi,ütles,downing,ma
2 TRUE
et,kui,ta,ning,suurbritannia,eli,juba,valimised,juhul,ent,erakorralised,järgi,jo,nende,te~
>
```

Hii-ruut test

Eelnenud proportsioonide testiga enamvähem sarnaselt kasutatakse hii-ruut testi, õigemini `prop.test`-i saab pidada `chisq.test`-i erijuhtumiks, kus korraga mõõdetakse vaid kahte arvu või tulpa. Hii-ruut testil selliseid piiranguid pole. Alustuseks sissejuhatav väljamõeldud näide õunte peal.

Tabel selle kohta, milliseid õunu millisel päeval kui palju korjati

```
> ounad=read_csv("http://www.tlu.ee/~jaagup/andmed/muu/ounad/ounad_paevad_2.txt")
> ounad
# A tibble: 3 x 3
  ounasort    esmaspaev reede
  <chr>           <int> <int>
1 Antoonovka         80    40
2 Valge klaar        60    30
3 Liivika           100    50
```

Testi tulemus:

```
> ounad %>% select(-ounasort) %>% chisq.test()

        Pearson's Chi-squared test

data:  .
X-squared = 0, df = 2, p-value = 1
```

Kuna reedel korjati esmaspäevast lihtsalt poole vähem õunu, aga suhted õunasortide vahel jäid samaks, siis tõenäosus, et korjati sarnaselt ehk nullhüpoteesi tõenäosus on 100% - ehk siis pole põhjust kahtlustada erinevat korjamist eri päevadel.

Teises täites korjati reedel esmaspäevast tunduvalt vähem Antoonovkaid ja rohkem Valgeid klaare, selle peale teatab ka test, et korje päevadel on erinev, ehk siis tõenäosus, et korjati ühtmoodi ja kogemata tulid sellised väärtused on 10 astmel -16.

```
> ounad=read_csv("http://www.tlu.ee/~jaagup/andmed/muu/ounad/ounad_paevad_1.txt")
> ounad
# A tibble: 3 x 3
  ounasort    esmaspaev reede
  <chr>           <int> <int>
1 Antoonovka         80    10
2 Valge klaar        60   200
3 Liivika           100   100
> ounad %>% select(-ounasort) %>% chisq.test()

        Pearson's Chi-squared test

data:  .
X-squared = 122.91, df = 2, p-value < 2.2e-16
```

Sõnade sagedused vastavalt pikkusele

Sisendiks kolme teksti sõnad

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikkused_haalikud.txt")
```

Vastavalt loole loeme kokku, kui palju kusagil on kolme tähe pikkusi sõnu

```
> sonad %>% filter(sonapikkus==3) %>% group_by(lugu) %>% summarise(kogus=n())
# A tibble: 3 x 2
  lugu      kogus
  <chr>     <int>
1 hinnad       31
2 kungla        8
3 lambipirn    56
```

Võrdluseks juurde ka viie tähe pikkused sõnad

```
> sonad %>% filter(sonapikkus %in% c(3, 5)) %>% group_by(lugu, sonapikkus) %>%
summarise(kogus=n())
# A tibble: 6 x 3
# Groups:   lugu [?]
  lugu      sonapikkus kogus
  <chr>          <int> <int>
1 hinnad             3    31
2 hinnad             5    25
3 kungla             3     8
4 kungla             5    12
5 lambipirn          3    56
6 lambipirn          5   100
```

Tulemus spread-käsu abil laia tabelisse

```
> sonad %>% filter(sonapikkus %in% c(3, 5)) %>% group_by(lugu, sonapikkus) %>%
+   summarise(kogus=n()) %>% ungroup() %>% spread(sonapikkus, kogus)
# A tibble: 3 x 3
  lugu        `3`   `5`
  <chr>     <int> <int>
1 hinnad       31    25
2 kungla        8    12
3 lambipirn    56   100
```

Siia otsa saab juba hii-ruut testi rakendada. Enne eemaldades loo nime kui sõnalise tunnuse.

```
sonad %>% filter(sonapikkus %in% c(3, 5)) %>% group_by(lugu, sonapikkus) %>%
  summarise(kogus=n()) %>% ungroup() %>% spread(sonapikkus, kogus) %>%
  select(-lugu) %>% chisq.test()
```

Tulemuseks tõenäosus lugude ühesuguste sõnapikkuste kohta 4%, ehk siis 96% tõenäosusega võime võita, et kolme- ja viietäheliste sõnade sagedus lugudes on erinev.

Pearson's Chi-squared test

data: .
X-squared = 6.4614, df = 2, p-value = 0.03953

Harjutus

- Tehke läbi näited õuntega, vahetage andmeid, jälgige tulemusi
- Käivitage näide kolme- ja viietäheliste sõnade arvudega
- Loendage iga loo kohta lisaks 10 tähe pikkusi sõnu. Kui mõnel lool vastava pikkusega sõna puudub, siis pange selle loenduri väärtuseks 0. Näidake hii-ruut testi abil, kuivõrd kindel on, et lugude kaupa on 3-, 5- ja 10-täheliste sõnade arv erinev

Õunte algsed andmed:

```
> ounad
# A tibble: 3 x 3
  ounasort    esmaspaev reede
  <chr>           <dbl> <dbl>
1 Antoonovka         80    40
2 Valge klaar        60    30
3 Liivika           100    50
```

Vahetatud väärtus

```
> ounad[1, "reede"]=60
```

```
> ounad
# A tibble: 3 x 3
  ounasort    esmaspaev reede
  <chr>           <dbl> <dbl>
1 Antoonovka         80    60
2 Valge klaar        60    30
3 Liivika           100    50
```

Testi tulemus:

```
> ounad %>% select(-ounasort) %>% chisq.test()

        Pearson's Chi-squared test

data:  .
X-squared = 3.4467, df = 2, p-value = 0.1785
```

Omistamine rea ja veeru numbri kaudu:

```
ounad[1, 3]=65
> ounad
# A tibble: 3 x 3
  ounasort    esmaspaev reede
  <chr>           <dbl> <dbl>
1 Antoonovka         80    65
2 Valge klaar        60    30
3 Liivika           100    50
```

Nagu näha, siis reedese 40 õuna puhul oli sarnasuse (ehk nullhüpoteesi) tõenäosus 100%, 60 õuna puhul 18%. Kui õunu aga on 40 asemel 65, siis tõenäosus, et selline jaotus reedel kogemata juhtus vaid 8%

```
> ounad %>% select(-ounasort) %>% chisq.test()

        Pearson's Chi-squared test

data:  .
X-squared = 5.0865, df = 2, p-value = 0.07861
```

Harjutuses siis esimest korda olukord, kus tulpasid rohkem kui kaks - ehk prop.test-i käsklusest ei piisaks. Mugavamaks vaatamiseks sõnapikkuse tulbale p-täht ette, nii ei teki hiljem arvulisi tulbanimesid, mille käsitsemine veidi tülikam

```
sonad %>% filter(sonapikkus %in% c(3, 5, 10)) %>% group_by(lugu, sonapikkus) %>%
  summarise(kogus=n()) %>% ungroup() %>% mutate(sonapikkus=paste("p",sonapikkus, sep=""))

# A tibble: 8 x 3
  lugu      sonapikkus kogus
  <chr>     <chr>      <int>
1 hinnad    p3            31
2 hinnad    p5            25
3 hinnad    p10           11
4 kungla    p3             8
5 kungla    p5            12
```

```
6 lambipirn p3            56
7 lambipirn p5           100
8 lambipirn p10           26
```

Andmed laia tabelisse

```
sonad %>% filter(sonapikkus %in% c(3, 5, 10)) %>% group_by(lugu, sonapikkus) %>%
  summarise(kogus=n()) %>% ungroup() %>% mutate(sonapikkus=paste("p",sonapikkus, sep="")) 
%>%
    spread(sonapikkus, kogus, fill=0)


  lugu        p10    p3    p5
  <chr>     <dbl> <dbl> <dbl>
1 hinnad       11    31    25
2 kungla        0     8    12
3 lambipirn    26    56   100
```

Hii-ruut testi tulemus

```
sonad %>% filter(sonapikkus %in% c(3, 5, 10)) %>% group_by(lugu, sonapikkus) %>%
  summarise(kogus=n()) %>% ungroup() %>% mutate(sonapikkus=paste("p",sonapikkus, sep="")) 
%>%
  spread(sonapikkus, kogus, fill=0) %>% select(-lugu) %>% chisq.test()


        Pearson's Chi-squared test

data:  .
X-squared = 9.9375, df = 4, p-value = 0.04149
```

Paistab, et kaasates ka 10 tähe pikkused sõnad võib endiselt ligikaudu 96% tõenäosusega väita, et sõnade pikkused lugudes on erinevad.

T-test

Aritmeetiliste keskmiste võrdlemine

Ühekordse arvutusena saame vastuse parajasti kättesaadavate andmete põhjal. Kui mõõdetakse mitmel korral ja (suhteliselt) juhuslikult kättesaadavate andmetega, siis hakkab aritmeetiline keskmine mõnevõrra kõikuma katsete vahel. Järgnevas näiteks Kungla rahva kümne juhusliku sõna pikkuste aritmeetiline keskmine

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
> sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus))
```

```
# A tibble: 1 x 1
      k
  <dbl>
1   4.1
> sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus))
# A tibble: 1 x 1
      k
  <dbl>
1   4.9
> sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus))
# A tibble: 1 x 1
      k
  <dbl>
1   4.5
```

Nagu näha, siis siin tuli vastuseks igal korral tabel, millel üks rida ja üks veerg. Palja väärtuse jaoks tuleb see sealt eraldi dollari ja veerunime järgi välja küsida.

```
> sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus)) %>%
.$k
[1] 4.3
```

Tahtes väärtusi suuremas koguses leida, aitab meid eelnevalt tuttav funktsioon sapply. Esialgu kümme katset

```
sapply(1:10, function(x){
  sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus)) %>%
.$k
})
```

```
[1] 5.4 4.3 4.6 3.7 4.3 4.3 6.0 5.8 4.4 4.3
```

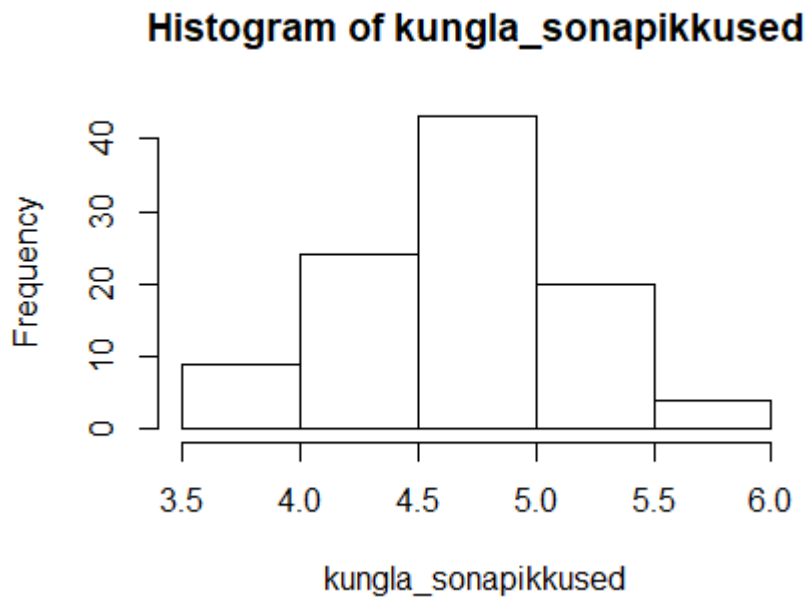
Suurema pildi jaoks sada

```
kungla_sonapikkused <- sapply(1:100, function(x){
  sonad %>% filter(lugu=="kungla") %>% sample_n(10) %>% summarise(k=mean(sonapikkus)) %>% .$k
})
```

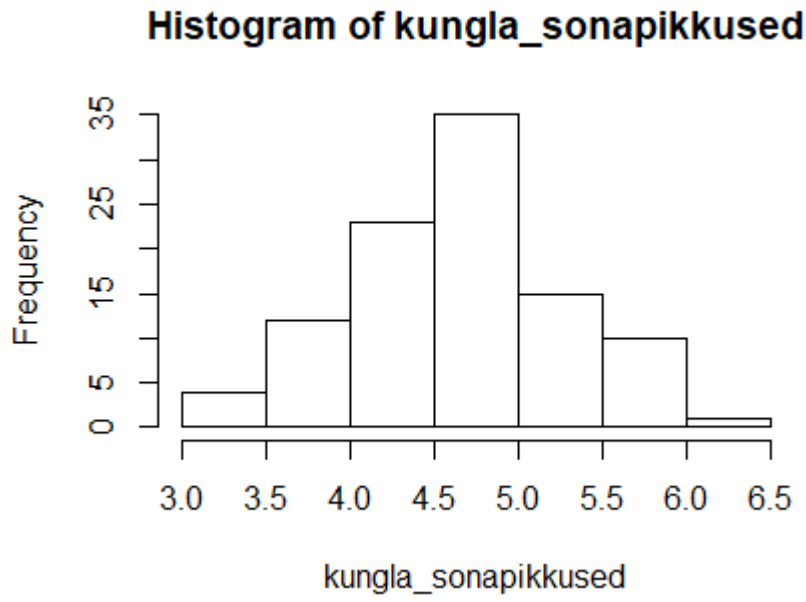
```
kungla_sonapikkused
 [1] 5.2 4.7 4.7 3.8 4.9 5.3 5.0 4.6 4.7 4.8 4.9 4.9 5.6 5.1 5.2 4.9 5.3 5.9 4.5 5.0 5.3 4.3 4.3
 [24] 5.1 5.2 5.0 5.7 4.5 4.5 4.8 4.0 4.6 5.3 5.2 5.2 4.2 4.4 5.3 4.6 4.5 4.7 3.5 4.5 3.5 4.7 5.0
 [47] 5.7 4.6 5.2 3.8 5.2 4.9 4.5 4.6 4.7 4.2 4.0 4.6 3.9 4.7 4.3 4.8 4.6 4.2 4.5 4.6 4.2 3.6 4.7
 [70] 4.9 4.0 4.5 5.3 4.5 5.2 4.8 5.3 4.5 4.9 5.0 4.9 4.9 5.1 4.8 4.7 4.9 4.7 4.7 5.1 4.5 4.3 4.8
 [93] 4.2 4.2 4.7 4.8 5.3 4.4 4.7 4.5
```

Andmete jaotusest ülevaate saamiseks historamm kümne kaupa võetud sõnade pikkuste keskmiste kohta

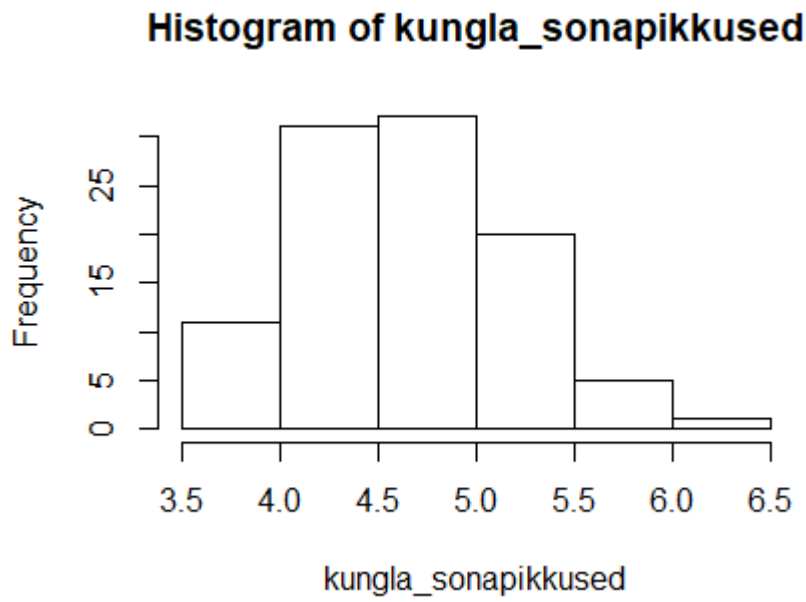
```
hist(kungla_sonapikkused)
```



Käskluse teist korda käivitamine annab mõnevõrra teravama keskmise tipuga jaotuse



Kolmandal korral kaldub tulemuste haripunkt sootuks veidi vasakule



Et kõigil juhtudel võti juhuslikud sõnad, siis saab kõiki neid jaotusi loomulikuks pidada. Lisaks ühtlasi näitavad välja, et katse ühel korral saadud tulemus võib teisel juhul vähemalt siin nähtud piirides erineda ilma, et sellest midagi üleloomulikku oleks.

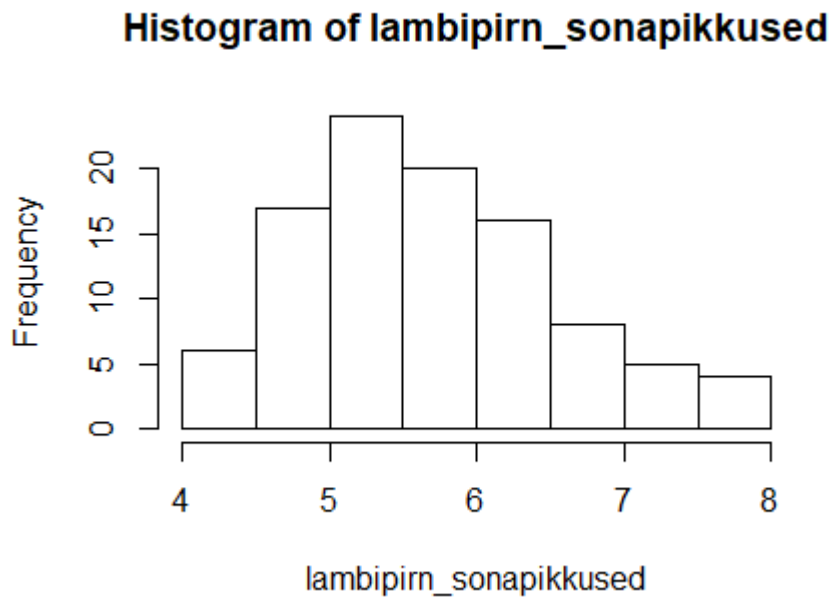
Harjutus

- Pange Kungla rahva juhuslike sõnade pikkuse histogrammi näide korduvalt käima. Katsetage korduste arvu väärtusi 10, 100 ja 1000
- Koosta sarnane histogramm Lambipirni loo sõnapikkuste kohta. Käivita korduvalt ning mitmesuguste korduste arvu väärtustega

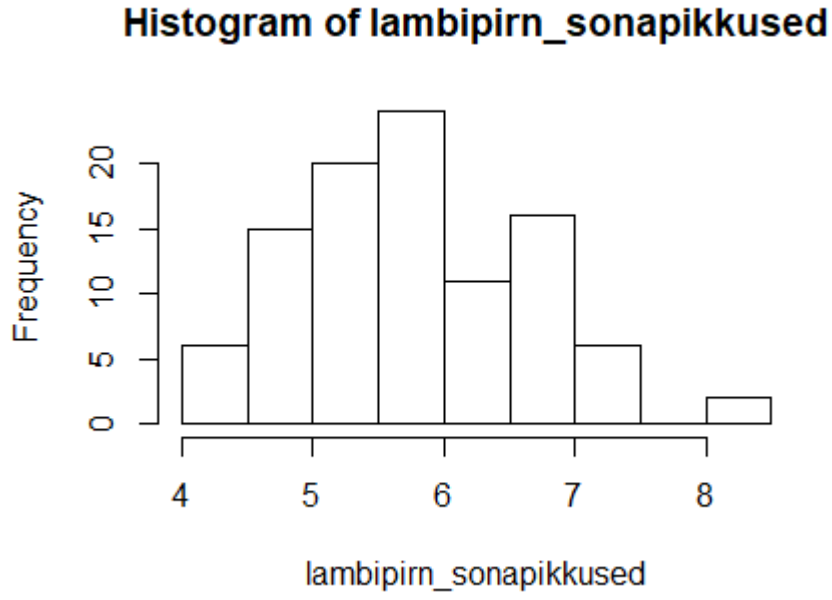
```
lambipirn_sonapikkused <- sapply(1:100, function(x){
  sonad %>% filter(lugu=="lambipirn") %>% sample_n(10) %>% summarise(k=mean(sonapikkus))
%>% .$k
})
```

```
lambipirn_sonapikkused
  [1] 5.3 6.4 5.3 5.4 7.6 5.4 5.6 7.2 4.9 4.9 6.8 4.0 5.5 6.3 6.3 7.3 5.1 5.3 6.0 6.1 5.2 4.9 5.8
 [24] 5.4 4.8 7.9 4.1 4.1 5.4 4.6 4.5 6.2 5.7 4.9 6.3 4.6 6.7 6.8 5.2 4.8 6.7 5.9 7.3 4.9 5.6 6.5
 [47] 5.8 6.4 5.8 4.7 7.5 5.4 6.1 5.4 5.6 6.3 5.8 5.0 5.7 6.5 6.0 6.8 4.6 6.1 4.4 5.1 5.9 6.8 4.7
 [70] 5.1 4.7 5.1 6.2 4.5 5.4 5.9 6.7 6.1 5.2 5.1 5.5 4.9 4.9 5.6 5.2 6.7 5.6 5.1 4.7 5.7 6.2 5.4
 [93] 6.0 5.9 7.1 6.2 7.7 5.7 5.4 7.7
```

```
hist(lambipirn_sonapikkused)
```



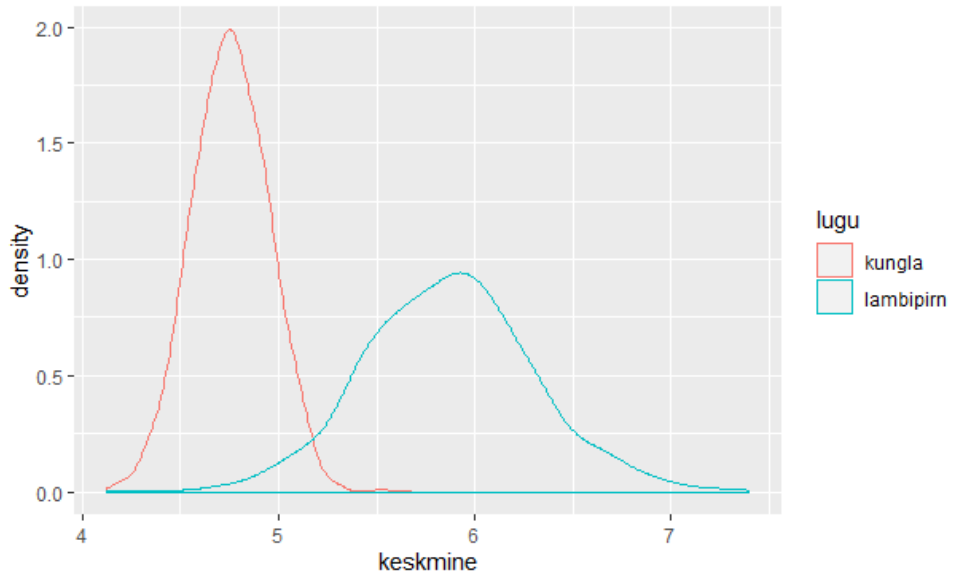
Teisel katsel tuleb Lambipirni teksti puhul olukord, kus mõned keskmised tulbad on madalamad kui tulbad nende kõrval. Jällegi täiesti loomulik juhtum, samas soovitatakse taolisel puhul teha joonis veidi väiksema arvuga jaotistega.



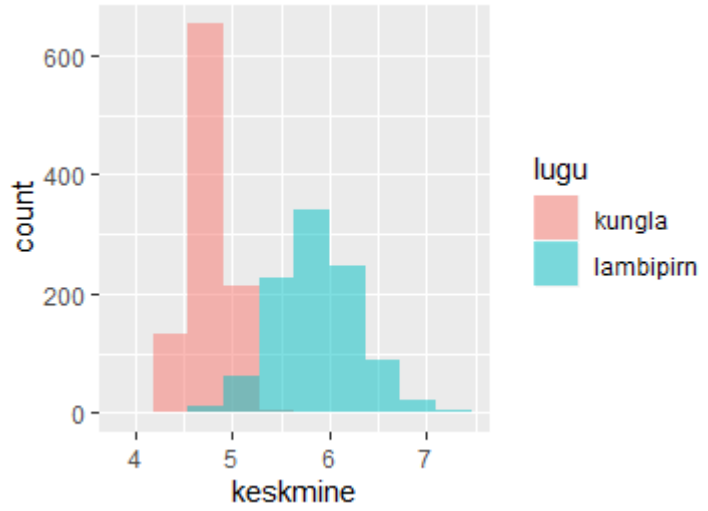
Mõlema loo keskmised sõnapikkused koos:

```
lambipirn_keskmised <- sapply(1:1000, function(x){
  sonad %>% filter(lugu=="lambipirn") %>% sample_n(40) %>% summarise(k=mean(sonapikkus))
%>% .$k
})

kungla_keskmised <- sapply(1:1000, function(x){
```

```
  sonad %>% filter(lugu=="kungla") %>% sample_n(40) %>% summarise(k=mean(sonapikkus)) %>%
.$k
})


keskmised <- tibble(kungla=kungla_keskmised, lambipirn=lambipirn_keskmised)
keskmised

# A tibble: 1,000 x 2
   kungla lambipirn
    <dbl>     <dbl>
 1   4.75      5.38
 2   4.72      4.7
 3   4.8       5.92
 4   4.62      5.98
 5   4.82      5.65
 6   4.15      5.5
 7   5.08      5.85
 8   4.97      5.72
 9   4.62      5.62
10   4.6       6.12
# ... with 990 more rows

> keskmised %>% gather(lugu, keskmine)
# A tibble: 2,000 x 2
   lugu   keskmine
   <chr>     <dbl>
 1 kungla     4.75
 2 kungla     4.72
 3 kungla     4.8
 4 kungla     4.62
 5 kungla     4.82
 6 kungla     4.15
 7 kungla     5.08
 8 kungla     4.97
 9 kungla     4.62
10 kungla     4.6
# ... with 1,990 more rows

keskmised %>% gather(lugu, keskmine) %>% ggplot(aes(keskmine, color=lugu)) + geom_density()
```



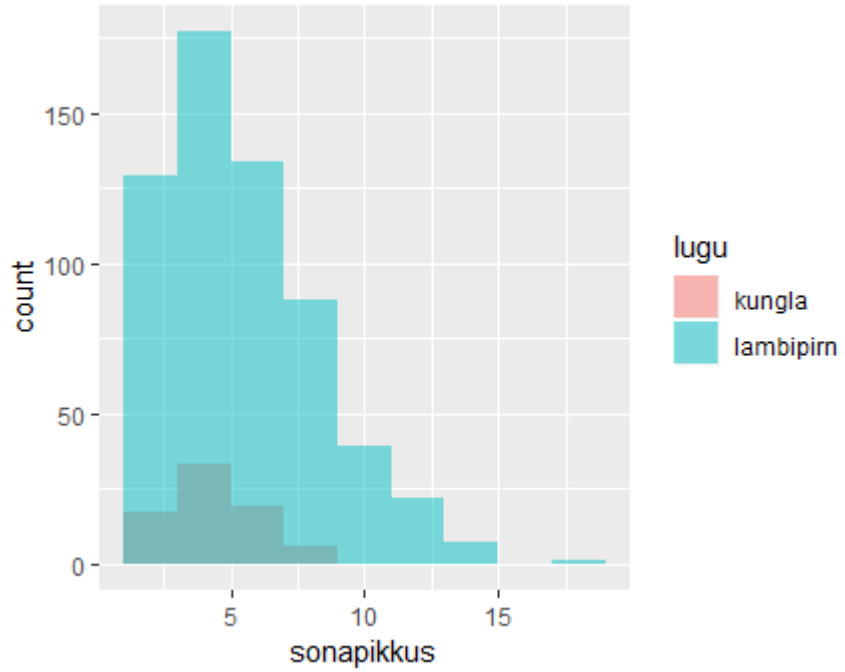
```
keskmised %>% gather(lugu, keskmine) %>% ggplot(aes(keskmine, fill=lugu)) +
geom_histogram(bins=10, alpha=0.5, position="identity")
```



Üksikute sõnade pikkused histogrammil

Mõlema teksti sõnapikkused samal histogrammil. Parameeter `fill=lugu` värvib kummagi loo sõnapikkused eri värvi, `position=identity` näitab, et kumbagi andmestikku tasub eraldi näidata; `binwidth=2` määrab tulba laiuse.

```
> sonad %>% ggplot(aes(sonapikkus, fill=lugu)) + geom_histogram(binwidth=2,
position="identity", alpha=0.5)
```



Kummastki loost 100 sõna

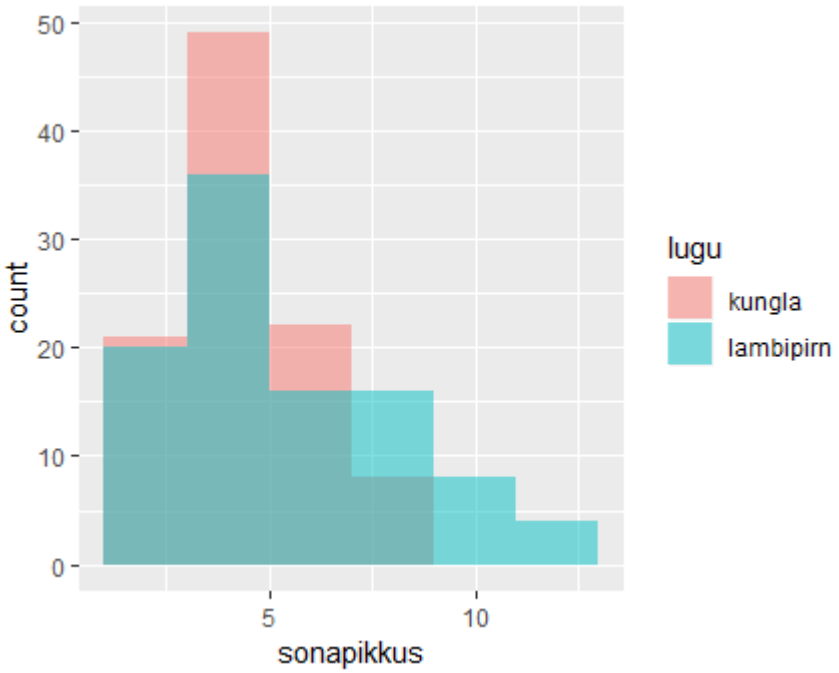
Eelmisel joonisel on näha, et Kungla rahva sõnu on märgatavalt vähem kui lambipirni omi. Üheks võrdsustamise võimaluseks on võtta kummasti loost ühepalju sõnu. Esimesel puhul on `replace=TRUE` tarvilik, sest Kungla rahva loos pole nõnda palju sõnu võtta ning mõnedel tuleb lubada korduda. Kahe tabeli read saab üheks lisada `bind_rows` käsu abil

```
sonad %>% filter(lugu=="kungla") %>% sample_n(100, replace=TRUE) %>% bind_rows(
  sonad %>% filter(lugu=="lambipirn") %>% sample_n(100))
```

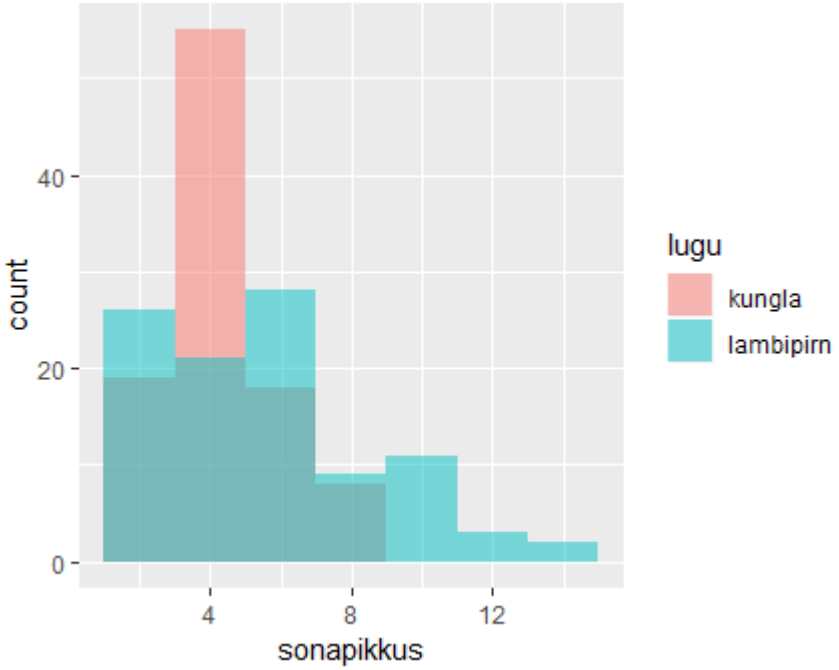
```
# A tibble: 200 x 5
   lugu   sona   sonapikkus taishaalikuid sulghaalikuid
   <chr>  <chr>       <int>         <int>         <int>
 1 kungla istus            5             2             1
 2 kungla laande           6             3             1
 3 kungla sööma            5             3             0
 4 kungla ja               2             1             0
 5 kungla ja               2             1             0
 6 kungla pähe             4             2             1
 7 kungla ja               2             1             0
 8 kungla siis             4             2             0
 9 kungla põksub           6             2             3
10 kungla loomad           6             3             1
# ... with 190 more rows
```

Sama käsklus ning tulemused saadetakse histogrammi loomiseks edasi. Läbipaistvus aitab üksteise peal olevatel tulpadel välja paista

```
sonad %>% filter(lugu=="kungla") %>% sample_n(100, replace=TRUE) %>% bind_rows(
  sonad %>% filter(lugu=="lambipirn") %>% sample_n(100)) %>%
  ggplot(aes(sonapikkus, fill=lugu)) +
    geom_histogram(binwidth=2, position="identity", alpha=0.5)
```



Mitme käivitamise puhul võib tulemus märgatavalt erineda, samas taas näha, et Kungla rahva sõnad rohkem ühte vahemikku koondunud.



Lugude esimeste sõnade võrdlemine, t-testi näide

Head-käsuga saab tabeli algused kätte nii ühe kui teise loo puhul.

```
> sonad %>% filter(lugu=="kungla") %>% head()
# A tibble: 6 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
4 kungla kuldsel          7             2             2
5 kungla aal              3             2             0
6 kungla kord             4             1             2

> sonad %>% filter(lugu=="lambipirn") %>% head()
# A tibble: 6 x 5
  lugu      sona          sonapikkus taishaalikuid sulghaalikuid
  <chr>     <chr>              <int>         <int>         <int>
1 lambipirn ehk                    3             1             1
2 lambipirn aitab                  5             3             2
3 lambipirn alljärgnev            10             3             1
4 lambipirn kallitel               8             3             2
5 lambipirn kodumaalastel         13             6             3
6 lambipirn vast                   4             1             1
```

Lihtsamal juhul need arvud t.test-i käsklusesse võrdlusse.

```
> t.test(c(3, 6, 6, 7, 3, 4), c(3, 5, 10, 8, 13, 4))
```

Tulemuste seletus:

Tõenäosus, et andmed on samast üldkogumist (p-value, nullhüpotees) on 0.219. Ehk siis ligi 80% on tõenäoline, et sõnapikkuste keskmine neis tekstides on erinev - juba kuue esimese sõna järgi. Allotsas näha kummagi loo algussõnade aritmeetiline keskmine pikkus (4,83 ja 7,17). Nende peal märgitud, et "95% tõenäosusega võime väita, et Kungla rahva loo sõnad on keskmiselt 6,43 tähte lühemad kuni 1,77 tähte pikemad". Kuna esialgu ainult kuut sõna vaadeldi, siis selline suhteliselt lai vahemik mõistetav.

```
        Welch Two Sample t-test

data:  c(3, 6, 6, 7, 3, 4) and c(3, 5, 10, 8, 13, 4)
t = -1.3497, df = 6.9072, p-value = 0.2197
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.432546  1.765879
sample estimates:
mean of x mean of y
 4.833333  7.166667
```

Terve loo sõnad ette võttes paistab vahe märgatavalt selgemini välja.

```
t.test(sonad %>% filter(lugu=="kungla") %>% .$sonapikkus,
       sonad %>% filter(lugu=="lambipirn") %>% .$sonapikkus)
```

Tõenäosus, et sõnapikkused võiksid tekstiti sarnased olla, on vaid 0,0000072, ehk 7,2*10^6 ehk 7.208e-06. Ülejäänud 0.999993 tõenäosusega järelikult on tekstide pikkuse aritmeetiline keskmine erinev. Õigemini kuna võrdlusse on võetud kogu tekstid, siis aritmeetiline keskmine tekstide vahel on mõõtmistulemuste järgi nagunii erinev. Test aga näitab, kuivõrd võib tulemust üldistada - juhul, kui kummalegi lisanduks sama tüüpi tekste. Välja on toodud kummagi teksti keskmine pikkus, samuti et Kungla rahva sõnad on 95% tõenäosusega lühemad 0,6 kuni 1,6 tähte.

```
        Welch Two Sample t-test

data:  sonad %>% filter(lugu == "kungla") %>% .$sonapikkus and sonad %>% filter(lugu ==
"lambipirn") %>% .$sonapikkus
t = -4.6823, df = 126.3, p-value = 7.208e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.6067994 -0.6520951
sample estimates:
mean of x mean of y
 4.760000  5.889447
```

Sama test oludes, kus kummastki loost võetakse 50 juhuslikku sõna

```
t.test(sonad %>% filter(lugu=="kungla") %>% sample_n(50) %>% .$sonapikkus,
       sonad %>% filter(lugu=="lambipirn") %>%  sample_n(50) %>% .$sonapikkus)

        Welch Two Sample t-test

data:  sonad %>% filter(lugu == "kungla") %>% sample_n(50) %>% .$sonapikkus and sonad %>%
filter(lugu == "lambipirn") %>% sample_n(50) %>% .$sonapikkus
t = -2.182, df = 79.725, p-value = 0.03205
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.06504226 -0.09495774
sample estimates:
mean of x mean of y
     4.54      5.62
```

Et sõnu vähem kui enne, siis usaldusvahemik laiem. Tõenäosus, et sõnade pikkused tekstides võiksid ühesugused olla on 3%, ehk siis erinevuse tõenäosus 97%

Sarnaselt proportsioonide testile võtame ka siin juhuslikud lähteandmed korduvalt ning jälgime, kuidas tekkinud usaldusvahemikud jaotuvad. Testi tulemuse saab korjata omaette muutujasse

```
testivastus <- t.test(sonad %>% filter(lugu=="kungla") %>% sample_n(50) %>% .$sonapikkus,
                      sonad %>% filter(lugu=="lambipirn") %>%  sample_n(50) %>%
.$sonapikkus)

testivastus
        Welch Two Sample t-test
```

```
data:  sonad %>% filter(lugu == "kungla") %>% sample_n(50) %>% .$sonapikkus and sonad %>%
filter(lugu == "lambipirn") %>% sample_n(50) %>% .$sonapikkus
t = -1.6033, df = 91.647, p-value = 0.1123
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.6119407  0.1719407
sample estimates:
mean of x mean of y
     5.00      5.72
```

Sealt saab edasi kätte usaldusintervalli

```
testivastus$conf.int
> testivastus$conf.int
[1] -1.6119407  0.1719407
attr(,"conf.level")
[1] 0.95
```

Hulgi andmete püüdmiseks sobib taas sapply käsklus.

```
testivastused <- sapply(1:10, function(x){
  testivastus <- t.test(sonad %>% filter(lugu=="kungla") %>% sample_n(50) %>% .$sonapikkus,
                        sonad %>% filter(lugu=="lambipirn") %>%  sample_n(50) %>%
.$sonapikkus)
  c(alates=testivastus$conf.int[1], kuni=testivastus$conf.int[2])
})
```

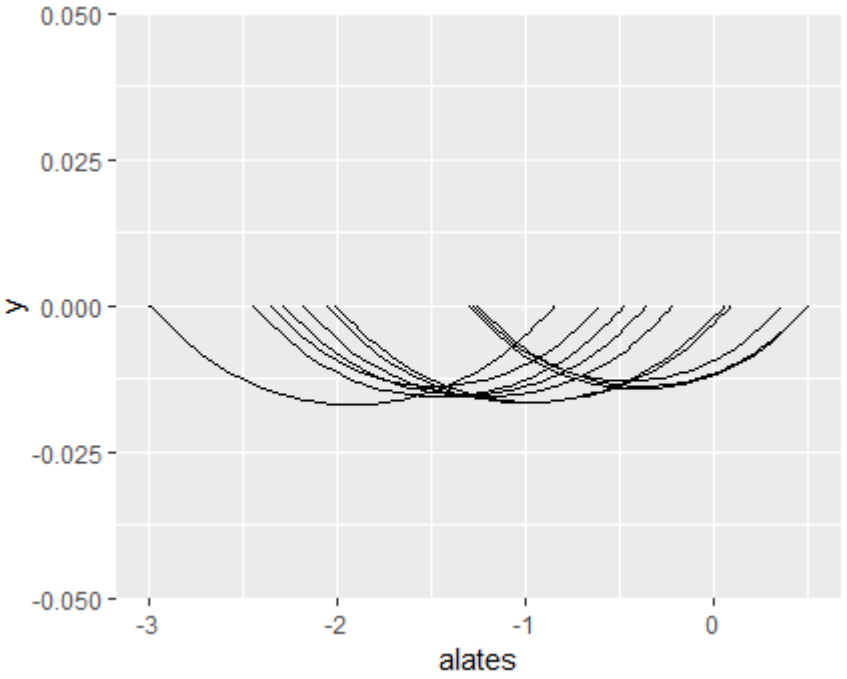
Hiljem t- käsklus (ehk transponse) vahetab read ja veerud, et tulemusi oleks mugavam joonisele kanda

```
t(testivastused)
        alates        kuni
 [1,] -2.993204 -0.84679600
 [2,] -2.352419 -0.60758136
 [3,] -1.304075  0.50407549
 [4,] -2.183304 -0.21669610
 [5,] -2.446662 -0.47333790
 [6,] -1.253420  0.49341989
 [7,] -2.007719  0.08771916
 [8,] -2.284946 -0.35505397
 [9,] -1.277997  0.35799688
[10,] -2.056415  0.05641494
```

Joonise loomine. Y-telje väärtused jätame kaare otspunktidel nullile, x-i väärtusteks tulevad tabeli `alates` ja `kuni`-tulpade väärtused.

```
ggplot(as_tibble(t(testivastused))) + geom_curve(aes(x=alates, y=0, xend=kuni, yend=0))
```

Jooniselt paistab, et mõnede kaarte otspunktid lähevad ka üle nulli piiri - ehk siis viiekümnesõnaliste näidete puhul ei saa osa juhtudel välistada, et andmed pakuvad ka võimalust, kus sõnade keskmised pikkused tekstides oleksid võrdsed. Samas üldpilt koondub ikkagi ligikaudu sinna sõnapikkuse ühetähelise erinevuse juurde.



Harjutus

- Käivita näide sõnapikkuste keskmiste võrdlemiseks T-testiga kummagi loo sõnades
- Kävita näide täishäälikute arvudega sõnades
- Leia iga sõna kohta täishäälikute osakaal (taishaalikuid/sonapikkus). Vihje: kui sõnas on kolm tähte, millest kaks täishäälikud, siis nende osakaal on ⅔ ehk 0,67
- Võrdle T-testi abil täishäälikute osakaalu (suhtelist sagedust) kummagi teksti sõnades

Lahendus

Kõigepealt uus tulp täishäälikute osakaaluga.

```
sonad2 <- sonad %>% mutate(tosakaal=taishaalikuid/sonapikkus)
sonad2
```

```
# A tibble: 672 x 6
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid tosakaal
   <chr>  <chr>        <int>         <int>         <int>    <dbl>
 1 kungla kui              3             2             1    0.667
 2 kungla kungla           6             2             2    0.333
 3 kungla rahvas           6             2             0    0.333
 4 kungla kuldsel          7             2             2    0.286
 5 kungla aal              3             2             0    0.667
 6 kungla kord             4             1             2    0.25
 7 kungla istus            5             2             1    0.4
 8 kungla maha             4             2             0    0.5
 9 kungla sööma            5             3             0    0.6
10 kungla siis             4             2             0    0.5
```

Edasi saab võrrelda kummagi loo juures osakaalu.

```
t.test(sonad2 %>% filter(lugu=="kungla") %>% .$tosakaal,
       sonad2 %>% filter(lugu=="lambipirn") %>% .$tosakaal)


       Welch Two Sample t-test

data:  sonad2 %>% filter(lugu == "kungla") %>% .$tosakaal and sonad2 %>% filter(lugu ==
"lambipirn") %>% .$tosakaal
t = 0.86435, df = 98.263, p-value = 0.3895
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01673102  0.04255392
sample estimates:
mean of x mean of y
0.4824392 0.4695277
```

Väljastatakse keskmised. Näha on, et Kungla rahva loos on täishäälikuid küll keskmiselt ühe protsendipunkti jagu rohkem, kuid sellest ei saa veel üldistavaid järeldusi teha.

Paarikaupa T-test

T-testiga arvukogumite keskväärtusi ehk aritmeetilisi keskmisi võrreldes võivad andmed tulla üldkogumist juhuslikult - praegusel korral sõnade tabelist viiekümne juhusliku sõna täishäälikute arv võrrelduna neljakümne juhusliku sõna sulghäälikute arvuga

```
> t.test(sonad %>% sample_n(50) %>% .$taishaalikuid,
         sonad %>% sample_n(40) %>% .$sulghaalikuid)

       Welch Two Sample t-test

data:  sonad %>% sample_n(50) %>% .$taishaalikuid and sonad %>% sample_n(40) %>%
.$sulghaalikuid
t = 4.5433, df = 87.363, p-value = 1.766e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.6947473 1.7752527
sample estimates:
mean of x mean of y
    2.760     1.525
```

Kui ka tegemist samade sõnadega, siis testikäsklus ei tea iseenesest ka selle samasusega arvestada

```
> t.test(sonad$taishaalikuid, sonad$sulghaalikuid)

       Welch Two Sample t-test
```

```
data:  sonad$taishaalikuid and sonad$sulghaalikuid
t = 21.435, df = 1310.2, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.247803 1.499220
sample estimates:
mean of x mean of y
 2.622024  1.248512
```

Testile on aga võimalik anda lisaparameeter `paired=TRUE`, sellisel puhul algoritm saab arvestada, et tegemist järjekorranumbri järgi samade objektidega, mille omadusi võrreldakse.

```
> t.test(sonad$taishaalikuid, sonad$sulghaalikuid, paired=TRUE)

        Paired t-test

data:  sonad$taishaalikuid and sonad$sulghaalikuid
t = 32.106, df = 671, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.289512 1.457512
sample estimates:
mean of the differences
               1.373512
```

Eelmises näites tuli 95% usaldusnivoo juures keskmiste erinevuse usaldusintervalliks 1,25 kuni 1,5; paarikaupa samasust arvestades 1,29 kuni 1,46 - ehk siis sama valimi juures veidi täpsem tulemus.

Harjutus

- Võrrelge täis- ja sulghäälikute arvu sõnas Kungla rahva tekstis eraldi sealt valitud 50 sõna põhjal. Näita tulemusi tavalise ning paarikaupa T-testi korral.
- Võrrelge täis- ja sulghäälikute osakaalu vahet Kungla rahva laulust valitud 50 sõna põhjal. Näita tulemusi tavalise ning paarikaupa T-testi korral.
- Leia samad andmed Lambipirni loo puhul
- Leia samad andmed lugude täistekste kasutades - kuivõrd muutuvad usaldusvahemikud

```
kungla50 <- sonad %>% filter(lugu=="kungla") %>% sample_n(50)
head(kungla50)
t.test(kungla50$taishaalikuid, kungla50$sulghaalikuid)
t.test(kungla50$taishaalikuid, kungla50$sulghaalikuid, paired=TRUE)
```

```
> kungla50 <- sonad %>% filter(lugu=="kungla") %>% sample_n(50)
> head(kungla50)
```

```
# A tibble: 6 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla maha             4             2             0
2 kungla laulan           6             3             0
3 kungla mets             4             1             1
4 kungla istus            5             2             1
5 kungla siis             4             2             0
6 kungla lehepuu          7             4             1
>
> t.test(kungla50$taishaalikuid, kungla50$sulghaalikuid)

        Welch Two Sample t-test

data:  kungla50$taishaalikuid and kungla50$sulghaalikuid
t = 8.9461, df = 88.298, p-value = 5.162e-14
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.306823 2.053177
sample estimates:
mean of x mean of y
     2.36      0.68


>
> t.test(kungla50$taishaalikuid, kungla50$sulghaalikuid, paired=TRUE)

        Paired t-test

data:  kungla50$taishaalikuid and kungla50$sulghaalikuid
t = 8.7228, df = 49, p-value = 1.532e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.29296 2.06704
sample estimates:
mean of the differences
                   1.68
```

Nagu näha, siis siin puhul paarikaupa võrdlus täpsuses võitu ei andnud

Ühepoolne T-test

Mõnel puhul on teada, et üks väärtus paratamatult ei saa teisest suurem olla, vaid paratamatult on võrdne või väiksem. Näiteks sõna täishäälikutest tähtede arv ei saa kuidagi ületada sõna kogu tähtede arvu. Seda saab ka T-test oma algoritmi juures arvestada.

Kõigepealt meeldetuletuseks tavaline kahepoolne T-test, kus arvuti andmetest midagi ei tea ning ta alles katse käigus avastab, et kumb väärtus suurem on.

```
> t.test(sonad$sonapikkus, sonad$taishaalikuid)

        Welch Two Sample t-test
```

```
data:  sonad$sonapikkus and sonad$taishaalikuid
t = 26.464, df = 932.07, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.908416 3.374322
sample estimates:
mean of x mean of y
 5.763393  2.622024
```

Nüüd samade andmete puhul T-test, kus käsule antakse märku, et esimesed väärtused saavad ainult suuremad (alternatiivhüpotees nullhüpoteesi ehk keskmiste võrdsuse asemel) olla. Kui ennist märgiti, et 95% tõenäosusega ületab sõnapikkuse keskmine täishäälikute arvu keskmist 2,91 kuni 3,37 tähte, siis teades, et teistpoolne suund pole võimalik, näidatakse, et sõnapikkus on keskmiselt vähemalt 2,95 tähe jagu suurem.

```
> t.test(sonad$sonapikkus, sonad$taishaalikuid, alternative = "greater")

        Welch Two Sample t-test

data:  sonad$sonapikkus and sonad$taishaalikuid
t = 26.464, df = 932.07, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 2.945928      Inf
sample estimates:
mean of x mean of y
 5.763393  2.622024
```

Sinna omakorda saab juurde märkida, et võrreldakse järgemööda samu sõnu, mis tõstab 95% tõenäosusega kinnitatud vahe veel kaugemale, 3,01 tähemärgi peale.

```
> t.test(sonad$sonapikkus, sonad$taishaalikuid, alternative = "greater", paired=TRUE)

        Paired t-test

data:  sonad$sonapikkus and sonad$taishaalikuid
t = 46.414, df = 671, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 3.029888      Inf
sample estimates:
mean of the differences
               3.141369
```

Harjutus

- Tehke näide läbi
- Võrrelge sarnaselt kahel moel sõna sulghäälikute arvu keskmist sõnapikkuse keskmisega

Võrdlus arvuga

Mõnikord on varasemast teada väärtus ning tahetakse t-testiga näidata, kas praeguse andmestiku aritmeetiline keskmine sellest üldistatavalt erineb. Selleks aitab t-testi juhtum, kus kogumit võrreldakse arvuga.

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
#Kas lambipirni sõnade keskmine pikkus võiks olla 5 tähte
t.test(sonad %>% filter(lugu=="lambipirn") %>% .$sonapikkus, mu=5)
```

Vastus:

Sõnade keskmine pikkus lambipirni loos on 95% tõenäosusega 5,7 kuni 6,1 tähte, järelikult sõnade keskmine pikkus ei saa olla 5 tähte.

```
One Sample t-test

data:  sonad %>% filter(lugu == "lambipirn") %>% .$sonapikkus
t = 7.5389, df = 596, p-value = 1.773e-13
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 5.657739 6.121156
sample estimates:
mean of x
 5.889447
```

Kungla rahva puhul jääb aga 5 uuritavasse vahemikku ning 26% tõenäosusega võib sõnade keskmine pikkus olla viis tähte

```
> t.test(sonad %>% filter(lugu=="kungla") %>% .$sonapikkus, mu=5)

        One Sample t-test

data:  sonad %>% filter(lugu == "kungla") %>% .$sonapikkus
t = -1.1407, df = 74, p-value = 0.2577
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 4.340776 5.179224
sample estimates:
mean of x
     4.76
```

Harjutus

- Näita lambipirni jutu juhusliku saja sõna põhjal, kui palju võiks sulghäälikute arv olla väiksem sõna pikkusest
- Näita lambipirni jutu juhusliku saja sõna põhjal, milline võiks olla sulghäälikute keskmine arv sõnas. Kas see võiks olla 1,5?

```
slambipirn = sample_n(sonad %>% filter(lugu=="lambipirn"), 100)
t.test(slambipirn$sulghaalikuid, slambipirn$sonapikkus, alternative="less", paired=TRUE)
```

```
        Paired t-test

data:  slambipirn$sulghaalikuid and slambipirn$sonapikkus
t = -20.235, df = 99, p-value < 2.2e-16
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -4.066491
sample estimates:
mean of the differences
                  -4.43
```

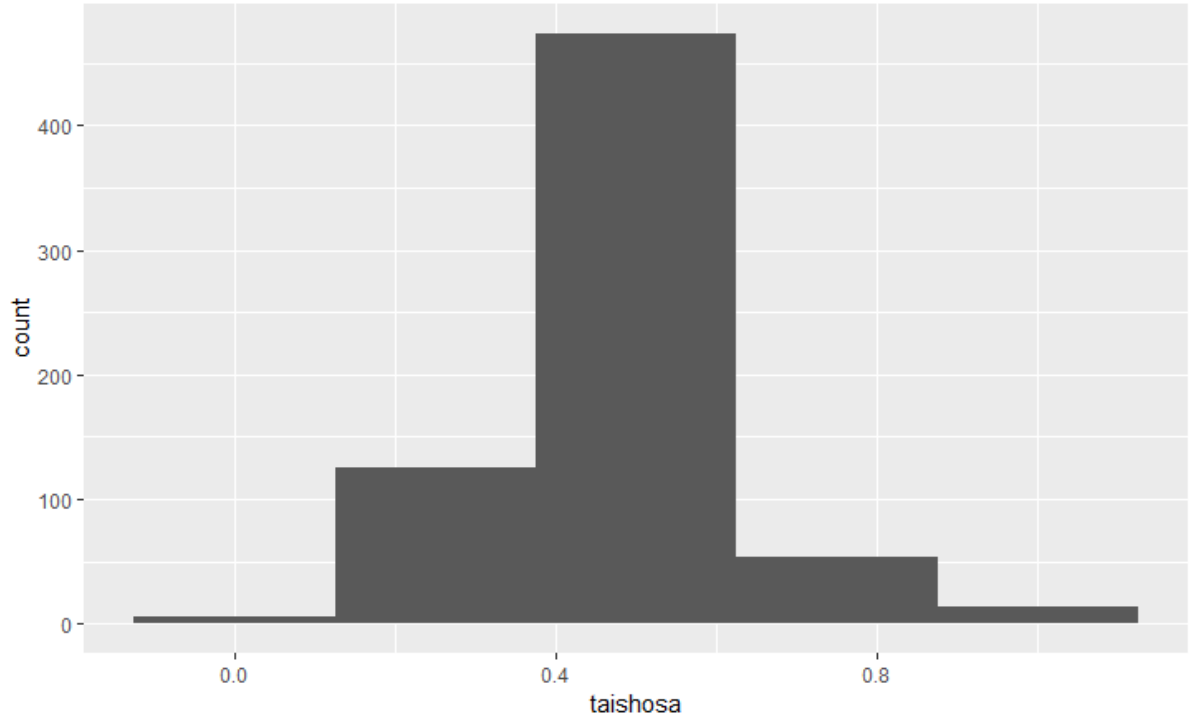
Jaotused

Matemaatikud on kokku pannud hulga jaotusi, mis seaduspärasustele sündmused eri puhkudel ja suunavate kõrvalmõjudeta olukordades jaotuvad. Väiksemate andmestike korral ei kooru jaotumine kuigi selgelt välja, heal juhul saame teoreetiliselt võrrelda ja arutleda, et andmed saaksid muid näitajaid arvestades ühe või teise jaotuse ja mingite parameetrite järgi jaotuda. Vaatlustulemustega võrreldes saab siis hinnata, et kas mõni mõõtmine on arutelu tulemustega märgatavalt vastuolus ning annab põhjuse tehtud katset ja selle tausta lähemalt kontrollida.

Normaaljaotus

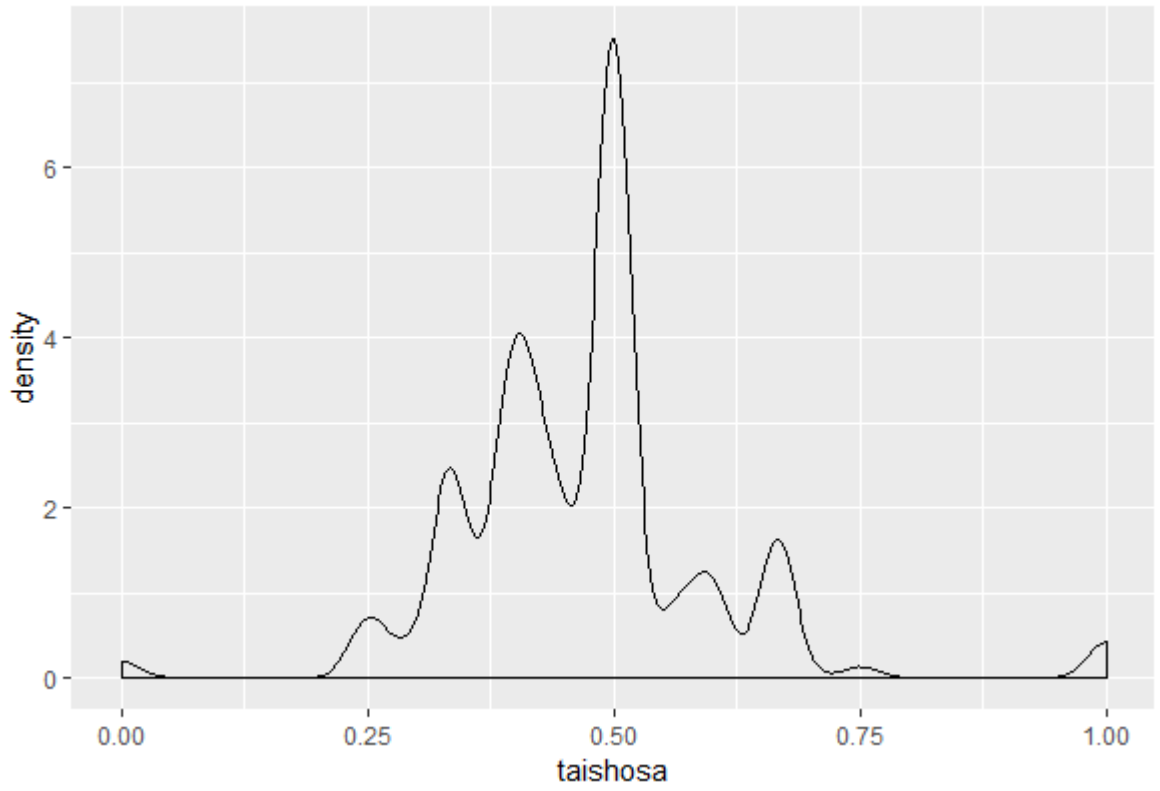
Näitena täishäälikute osakaal uuritud teksti sõnades

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
sonad  %>% mutate(taishosa=taishaalikuid/sonapikkus) %>%
  ggplot(aes(taishosa)) + geom_histogram(bins = 5)
```



Enamikel sõnadel paistavad ligikaudu pooled tähed olema täishäälikute omad. Käsu geom_density abil saab sagedusjaotuse küsida joonena

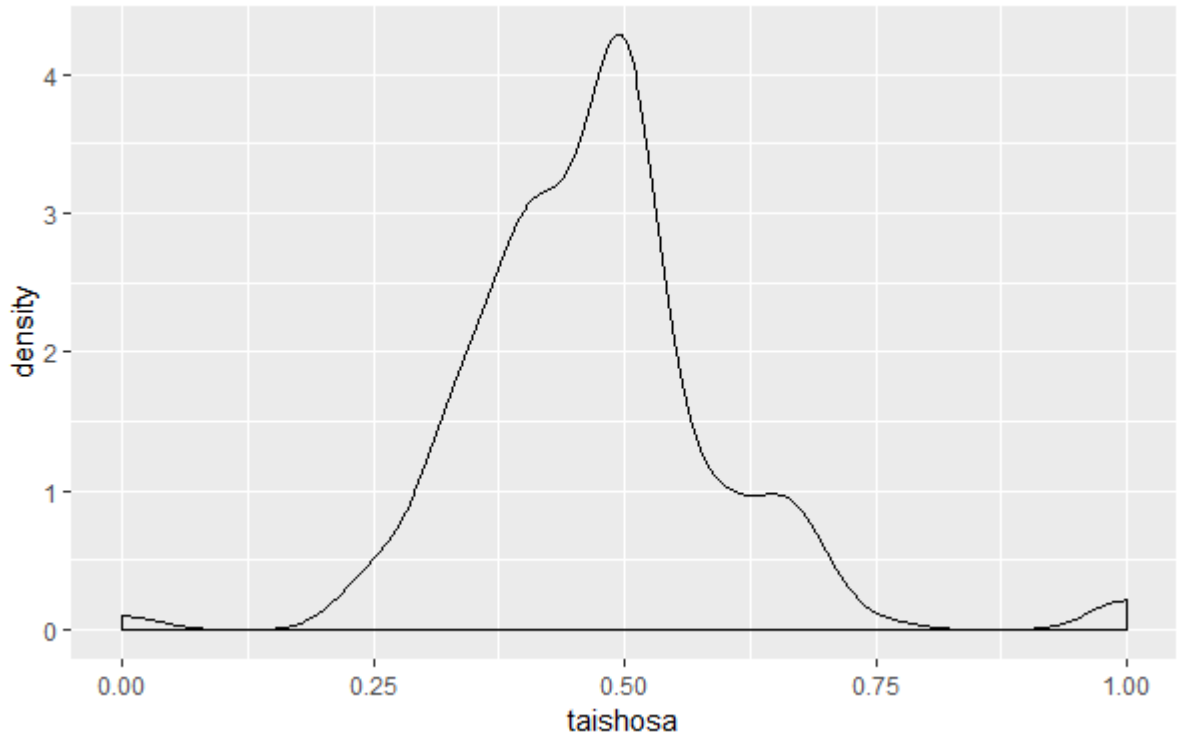
```
sonad  %>% mutate(taishosa=taishaalikuid/sonapikkus) %>%
  ggplot(aes(taishosa)) + geom_density()
```



Jällegi piik ülespoole kohas, kus pooled sõna tähtedest on täishäälikute omad. Kõikumised paistavad aga üpriski suured olema muuhulgas tõenäoliselt seetõttu, et tähti ja täishäälikuid on sõnas täisarvuline arv ning siis suur osa suhteid pole lihtsalt võimalikud.

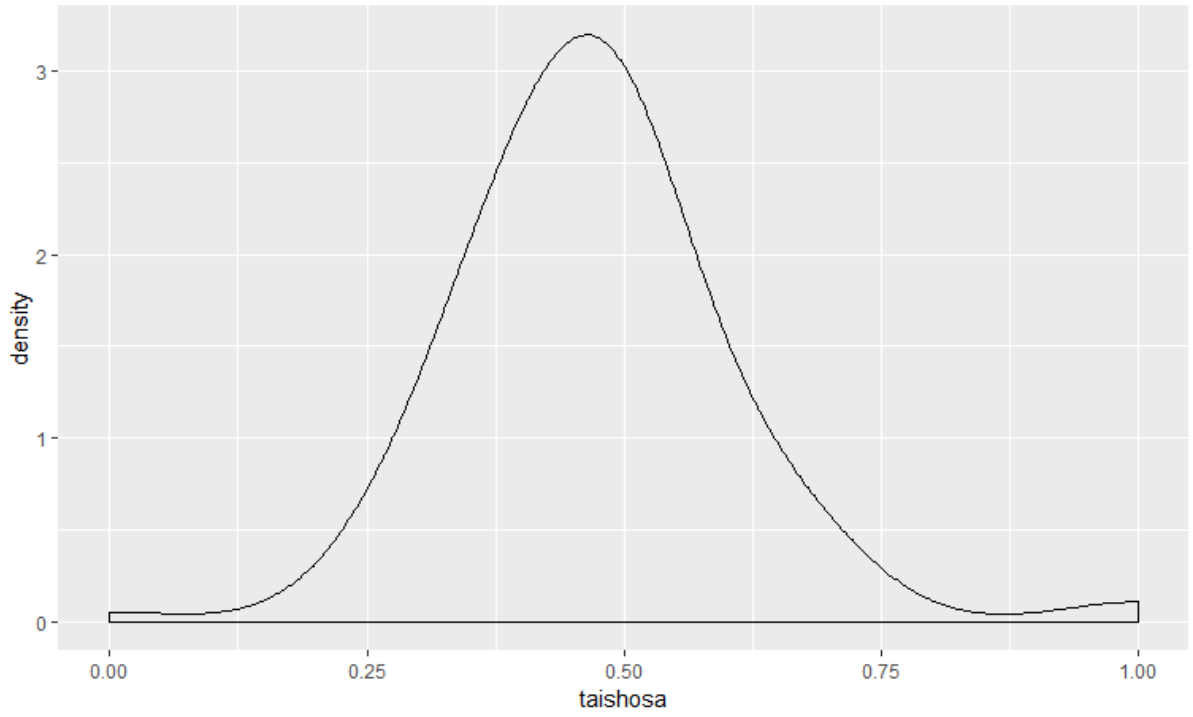
Graafiku saab sujuvamaks, muutes parameetri adjust väärtust. Kahekordne silendus teeb joone juba märgatavalt sujuvamaks

```
sonad  %>% mutate(taishosa=taishaalikuid/sonapikkus) %>%
     ggplot(aes(taishosa)) + geom_density(adjust=2)
```



Neljakordse silendamise puhul puuduvad järsud jõnksud täiesti. On aga näha, et Gaussi kõveraga võrreldes mõned erisused on selgelt sees - keskväärtus on küll 0,5 juures, aga leiduvad mõned sõnad, kus pole ühtegi täishäälikut - või siis koosnevad tervikuna täishäälikutest.

```
sonad  %>% mutate(taishosa=taishaalikuid/sonapikkus) %>%
    ggplot(aes(taishosa)) + geom_density(adjust=4)
```

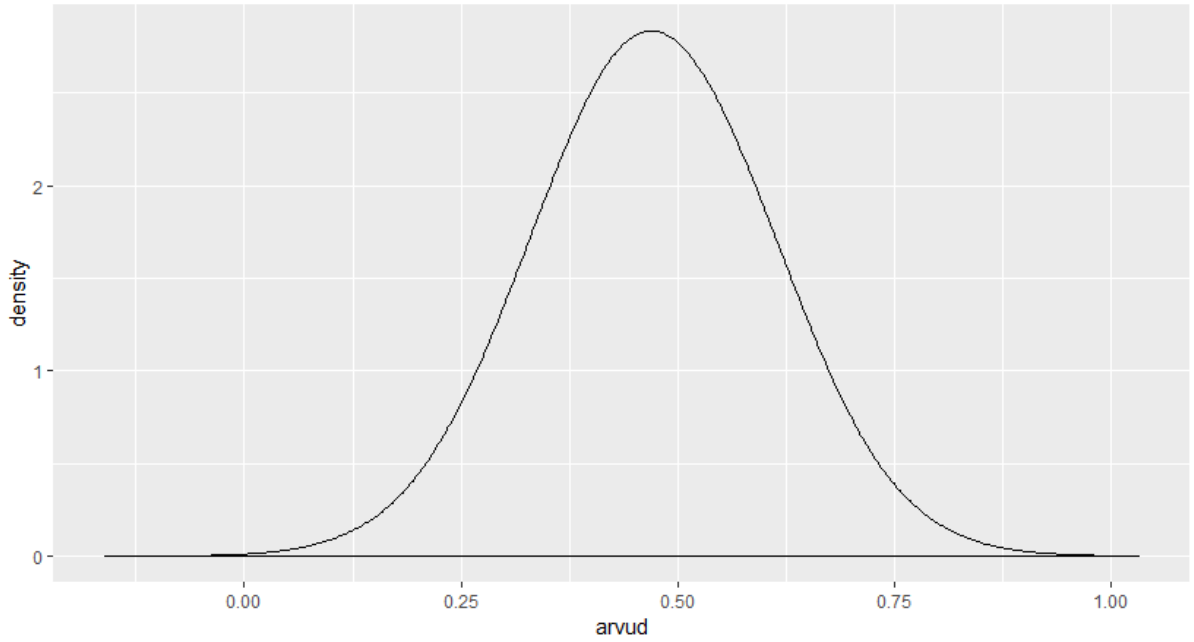


Leiame täishäälikute osakaalu aritmeetilise keskmise ning standardhälbe

```
sonad  %>% mutate(taishosa=taishaalikuid/sonapikkus) %>%
     summarise(kesk=mean(taishosa), shalve=sd(taishosa))
# A tibble: 1 x 2
   kesk shalve
  <dbl>  <dbl>
1 0.471  0.132
```

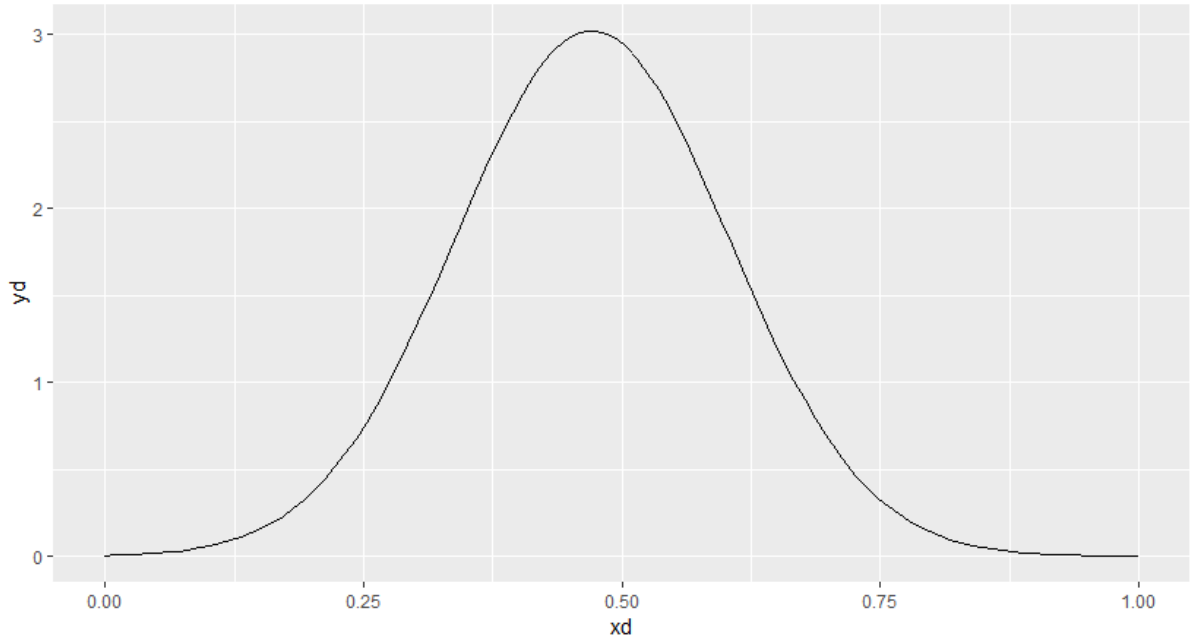
Laseme genereerida sada tuhat väärtust aritmeetilise keskmisega 0,471 ning standardhälbega 0,132 ning vaatame, kuidas nende tihedus jaotub

```
tibble(arvud=rnorm(100000, mean=0.471, sd=0.132)) %>% ggplot(aes(arvud)) +
geom_density(adjust=4)
```



Sarnase tulemuse saame ka, kui arvutame normaaljaotuse tihedusfunktsiooni oma mõõtmisest saadud aritmeetilise keskmise ja standardhälbe juures. Jooniselt paistab, et sõna puhul täishäälikute osakaal 0.5 lähedal on ligi neli korda tõenäolisem kui 0.25 juures.

```
tibble(xd=seq(0, 1, length=100)) %>%
 mutate(yd=dnorm(xd, mean=0.471, sd=0.132)) %>%
 ggplot(aes(xd, yd))+geom_line()
```



Sama joonise saab ka funktsiooni kuju välja joonistamiseks mõeldud käsu `stat_function` abil

```
tibble(xd=seq(0, 1, length=100)) %>%
  ggplot(aes(xd))+stat_function(fun=dnorm, args=c(mean=0.471, sd=0.132))
```

Küsime ka välja normaaljaotuse puhul tihedusfunktsiooni väärtused x-i ehk täishäälikute osakaalu eri väärtuste puhul

```
> dnorm(0.25, mean=0.471, sd=0.132)
[1] 0.7441378
> dnorm(0.5, mean=0.471, sd=0.132)
[1] 2.950225
> dnorm(0.75, mean=0.471, sd=0.132)
[1] 0.3237724
> dnorm(1.0, mean=0.471, sd=0.132)
[1] 0.0009835748
> dnorm(1.25, mean=0.471, sd=0.132)
[1] 8.270997e-08
```

Viimasest paistab, et normaaljaotus ei välista ka täishäälikute osakaalu rohkem kui 100% - ehk siis kui ainult sellele toetuda, siis on lubatud ka sõnad, milles oleks täishäälikulisi tähti rohkem kui üldse tähti kokku. Pealtnäha tundub, et tegemist on jamaga - kuid teistpidi tasub vaadata, et matemaatiline kõver on paratamatult ainult tegelikkuse lähend ning ainult täishäälikutest koosneva sõna tõenäosus on 1000 korda väiksem kui muu keskmine osakaal. Olukord, et täishäälikuid oleks tähtede arvust 25% jagu rohkem, on üks kaheksasaja miljoni kohta - ehk ligikaudu üks inimene hiinas või liivatera kuupmeetris - mis sõltuvalt mudeli kasutuskohast on loodetavasti enamikel juhtudel siiski piisav täpsus.

Millise väärtuseni on milline osa mõõtmistest - qnorm

Kui oleme suutnud oma nähtust iseloomustava kõvera mudeli luua, siis saab selle juurest üht-teist küsida. Käsu qnorm abil küsime, et kuni millise väärtuseni on poolte sõnade täishäälikute osakaal. Et normaaljaotus on sümmeetriline mõlemale poole, siis vastus tuleb arvatult aritmeetilise keskmise kohale - jaotuse järgi on poolte sõnade täishäälikute osakaal väiksem kui 0,471 - ehk siis saab samamoodi järeldada, et ülejäänud poolte sõnade täishäälikute osakaal on suurem kui 0,471.

```
> qnorm(0.5, mean=0.471, sd=0.132)
[1] 0.471
```

Järgmise küsimise peale saame vastuseks, et 10% sõnade täishäälikute osakaal jaotuse järgi on väiksem kui 0,302 - järelikult 90% sõnade puhul on täishäälikute osakaal jaotuse järgi suurem kui 0,302. Et peame normaaljaotuses kõiki väärtusi võimalikeks, siis täpselt võrdumisega ei arvesta.

```
> qnorm(0.1, mean=0.471, sd=0.132)
[1] 0.3018352
```

Veel väiksemaks minnes näitab jaotus, et vaid 1% sõnu võiks olla täishäälikute osakaaluga vähem kui 16%. Tegelikkus läheb siin näites lahku, sest sõnadeks on loetud ka lühendid ja arvud - nii et siin otsas enam andmestik jaotusele niivõrd ei allu.

```
> qnorm(0.01, mean=0.471, sd=0.132)
[1] 0.1639221
```

99% sõnu võiksid olla täishäälikute osakaaluga alla 78%

```
> qnorm(0.99, mean=0.471, sd=0.132)
[1] 0.7780779
```

Harjutus

- Leia, millisest väärtusest väiksem võiks olla täishäälikute osakaal 5% sõnadest, juhul, kui osakaal on jaotunud normaaljaotuse järgi
- Leia inimeste kogumi kohta, kelle pikkuse aritmeetiline keskmine on 170cm ja standardhälve 10cm normaaljaotuse järgi, millisest väärtusest väiksemad võiksid olla 10% inimesi, 1% inimesi. Millisest väärtusest väiksemad võiks olla 99,9% inimesi (kui kõrge bussi laega on mõtet arvestada)

```
> qnorm(0.999, 170, 10)
[1] 200.9023
```

Väärtuse järgi osa leidmine - pnorm

Teadaoleva normaaljaotuskõvera korral saab võrdluseks ka vastupidi küsida - anname ette täishäälikutähtede suurima osakaalu sõnas ning küsime, kui suurel osal sõnadest on kuni selline täishäälikute osakaal. Kontrolliks küsime keskväärtuse 0.471 järgi - ettearvatult on pooltel juhtudel sõnades vähem kui 0.471 osa täishäälikuid - ning järelikult normaaljaotuse järgi pooltel juhtudel jällegi rohkem kui 0.471 osa täishäälikuid.

```
pnorm(0.471, mean=0.471, sd=0.132)
[1] 0.5
```

Vaatame ka teisi arvutusi tagurpidi - sõnu, millel on täishäälikuid kuni 77,8%, on normaaljaotuse käest küsides 99%

```
> pnorm(0.7780779, mean=0.471, sd=0.132)
[1] 0.99
```

Uut paari vaadates - kümnel protsendil sõnades on normaaljaotuse järgi vähem kui 30% täishäälikuid

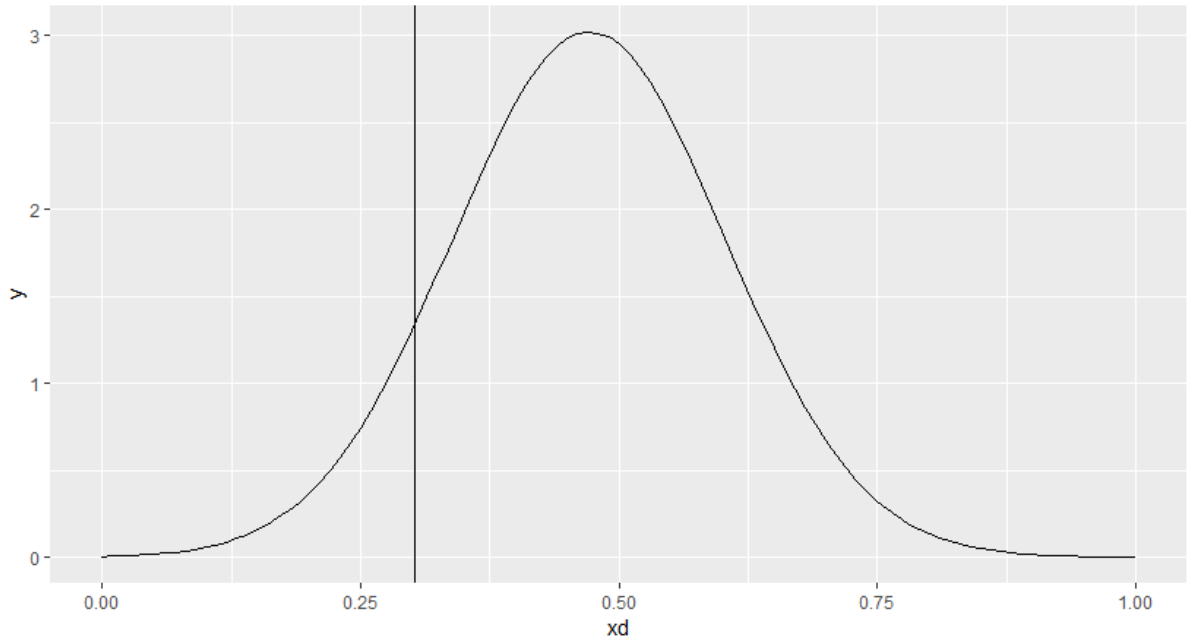
```
> qnorm(0.1, mean=0.471, sd=0.132)
[1] 0.3018352
```

Tagurpidi küsides - kuni 30% täishäälikuid on kümnel protsendil sõnadest.

```
> pnorm(0.3018352, mean=0.471, sd=0.132)
[1] 0.1
```

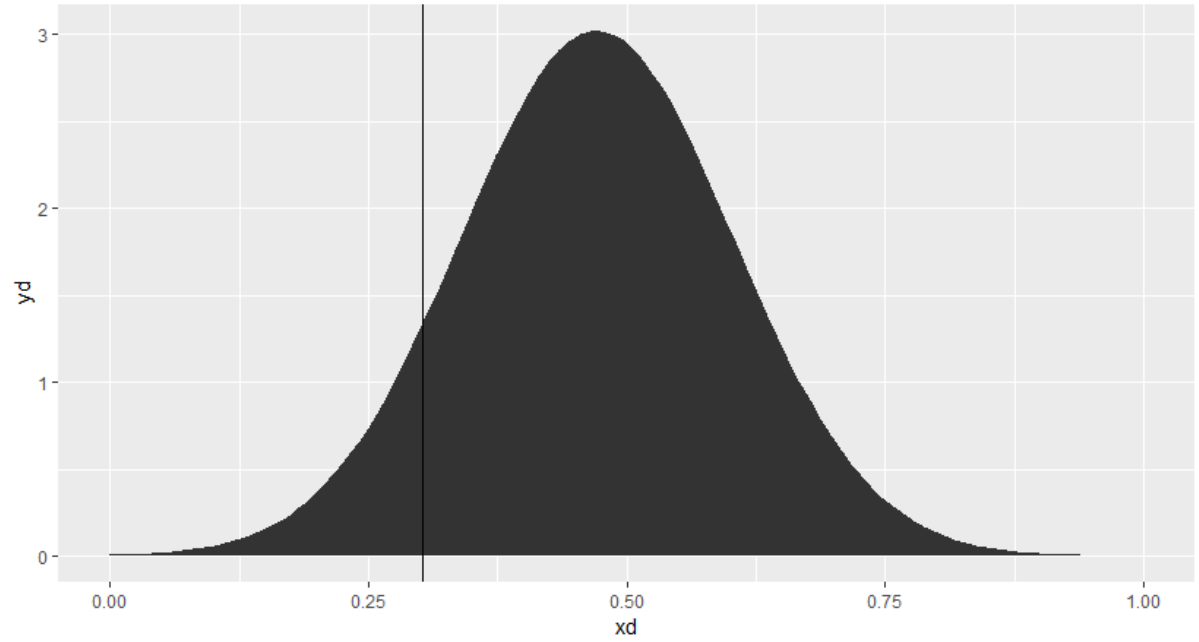
Nood kümme protsendi jäävad joonisel püstjoonest vasakule poole

```
tibble(xd=seq(0, 1, length=100)) %>%
  ggplot(aes(xd))+stat_function(fun=dnorm, args=c(mean=0.471, sd=0.132)) +
  geom_vline(xintercept=0.302)
```



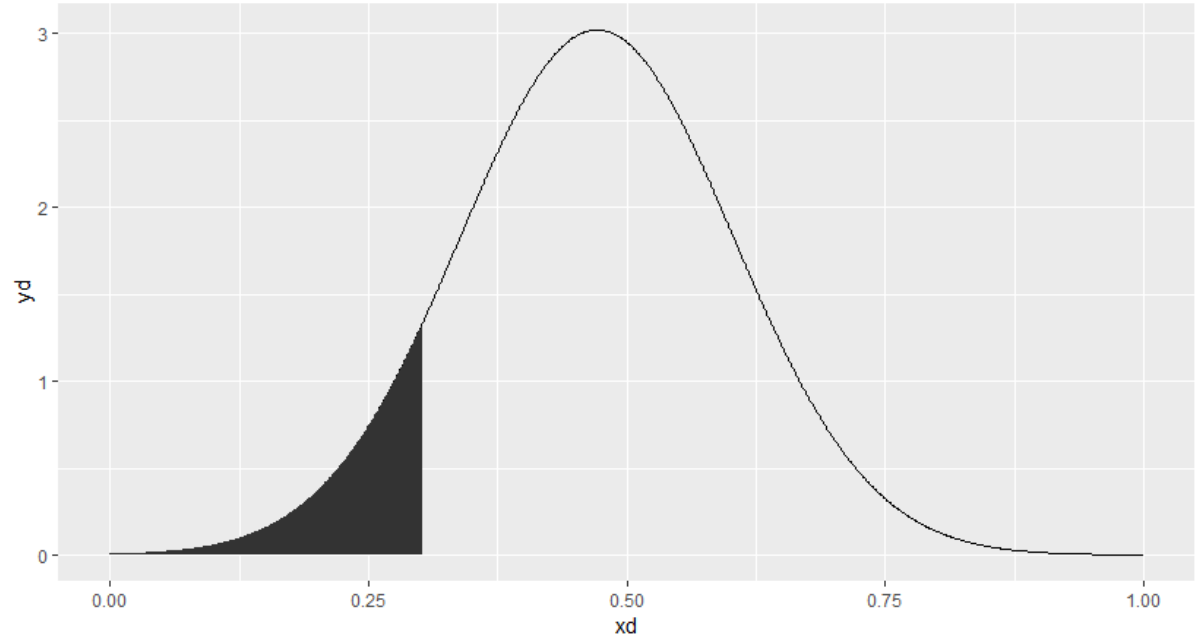
Vastavalt täishäälikute osakaalule sõnade esinemistõenäosus värvitud pinnana käsu geom_area abil

```
tibble(xd=seq(0, 1, length=100)) %>%
  mutate(yd=dnorm(xd, mean=0.471, sd=0.132)) %>%
  ggplot(aes(xd, yd))+geom_area() + geom_vline(xintercept=0.302)
```



Värvituna ainult kümnele protsendile pinnast/sõnadest vastav osa

```
tibble(xd=seq(0, 1, length=1000)) %>%
  mutate(yd=dnorm(xd, mean=0.471, sd=0.132)) %>%
  ggplot(aes(xd, yd))+geom_line() + geom_area(aes(y=ifelse(xd<=0.302, yd, 0)))
```



Harjutus

- Pane näited tööle
- Koosta Gaussi kõvera joonis inimeste pikkuste jaotumise kohta oludes, kus keskväärtus on 170 cm ning standardhälve 10 cm

- Leia pnorm-i abil, kui suur osa pikkustest on väiksem kui keskväärtus miinus üks standardhälve ehk praeguste andmete puhul väiksem kui 160 cm.
- Leia, kui suur osa pikkustest on eelnevas jaotuses väiksemad kui 180 cm.
- Leia, milline protsent pikkustest jääb aritmeetilise keskmise +/- ühe standardhälbe vahele (praegusel juhul 160 ja 180 cm vahele)
- Leia, milline protsent pikkustest jääb aritmeetilise keskmise +/- kahe standardhälbe vahele.
- Kuva 150 cm-st väiksemate pikkustele vastav pindala joonisel
- Kuva 160 cm ja 180 vahele jäävate pikkuste pindala joonisel.

Binoomjaotus

Neljatäheliste sõnade puhul saab täishäälikuid olla null kuni neli. Sellise kindlate väärtuste arvuga tegeleb binoomjaotus. Parameetriks on seeria pikkus ning sündmuse esinemise tõenäosus. Näitena neljatäheliste sõnade täishäälikuliste tähtede arvud: ilma täishäälikuta oli kaks sõna, ühe täishäälikuga 27 sõna, kahe täishäälikuga 86 sõna, kolme täishäälikuga 4 sõna ning nelja täishäälikuga polnud ühtegi sõna.

```
sonad %>% filter(sonapikkus==4) %>% group_by(taishaalikuid) %>% summarise(kogus=n())
# A tibble: 4 x 2
  taishaalikuid kogus
          <dbl> <int>
1             0     2
2             1    27
3             2    86
4             3     4
```

Leiame täishäälikute arvu, kõikide tähted arvu ning täishäälikuliste tähtede osakaalu

```
sonad %>% filter(sonapikkus==4) %>% summarise(tsum=sum(taishaalikuid),
psum=sum(sonapikkus), osakaal=tsum/psum)
# A tibble: 1 x 3
   tsum  psum osakaal
  <dbl> <dbl>   <dbl>
1   211   476   0.443
```

Sealtkaudu saame kätte binoomjaotuse andmed - seeria pikkus 4, täishääliku esinemise tõenäosus 0,443.

pbinom

Käsuga pbinom saab kontrollida, et kui suur osa andmetest peaks uuritava väärtuseni olema. Ehk siis ühegi täishäälikuta peaks olema 9.6% neljatähelistest sõnadest juhul, kui täishääliku esinemine on juhuslik.

```
> pbinom(0, 4, 0.443)
[1] 0.09625444
```

Nulli või ühe täishäälikuga 40,2%

```
> pbinom(1, 4, 0.443)
[1] 0.4024714
```

Kuni kahega kolmveerand sõnadest

```
> pbinom(2, 4, 0.443)
[1] 0.7677878
```

Kuni kolmega 96%

```
> pbinom(3, 4, 0.443)
[1] 0.9614863
```

Ja nagu aimata, siis 100% neljatähelistest sõnadest on kuni nelja täishäälikuga.

```
> pbinom(4, 4, 0.443)
[1] 1
```

Arvuti käest võib igasugu asju küsida - temale on iseenesestmõistetav ka see, et 100% neljatähelistest sõnadest täishääliku esinemissageduse 0,443 juures on kuni viie täishäälikuga (0-4 lähevad paratamatult sinna sisse)

```
> pbinom(5, 4, 0.443)
[1] 1
```

dbinom

Täpselt niimitme täishäälikuga eeldatavat sõnade arvu saab küsida `dbinom`-käsuga, eesliide sõnast *density*.

Sarnaselt eelmisel võiks jaotuse järgi arvestades olla täishäälikuta sõnu 9,6%

```
> dbinom(0, 4, 0.443)
[1] 0.09625444
```

Edasi aga andmed erinevad - ühe täishäälikuga võiks olla 30,6%.

```
> dbinom(1, 4, 0.443)
[1] 0.306217
```

Arvutades paistab, et nulli täishäälikuga sõnade teoreetiline osakaal + ühe täishäälikuga sõnade teoreetiline osakaal on sama kui kuni ühe täishäälikuga sõnade teoreetiline osakaal. Lihtsalt kumba kaudu on parasjagu kasulikum arvutada.

```
> dbinom(0, 4, 0.443)+dbinom(1, 4, 0.443)
[1] 0.4024714
> pbinom(1, 4, 0.443)
[1] 0.4024714
```

Edasi ka kahe, kolme ja nelja täishäälikulise tähega sõnade osakaalud

```
> dbinom(2, 4, 0.443)
[1] 0.3653163
> dbinom(3, 4, 0.443)
[1] 0.1936985
> dbinom(4, 4, 0.443)
[1] 0.03851367
```

Programm lubab küsida ka viie täishäälikuga sõnade osakaalu neljatäheliste sõnade hulgas - aga see on viisakasti null

```
> dbinom(5, 4, 0.443)
[1] 0
```

Jaotuste sarnasuste võrdlemine

Neljatäheliste sõnade hulgas on täishäälikuid kokku 211:

```
> sonad %>% filter(sonapikkus==4) %>% summarise(tsum=sum(taishaalikuid))
# A tibble: 1 x 1
   tsum
  <dbl>
1   211
```

Teoreetilise jaotuse korral oleks ühe täishäälikuga sõnu loetelus ligikaudu 20

```
> dbinom(0, 4, 0.443) * 211
[1] 20.30969
```

Iga täishäälikute arvu kohta tuleb

```
> sapply(0:4, function(nr){dbinom(nr, 4, 0.443)*211})
[1] 20.309687 64.611788 77.081747 40.870394  8.126384
```

Tegelikud saab tekstist välja korjata, teoreetilised kogused samuti muutujasse panna.

```
> tegelikud=sapply(0:4, function(nr){sonad %>% filter(sonapikkus==4, taishaalikuid==nr) %>%
nrow()})
> tegelikud
[1]  2 27 86  4  0
> teoreetilised=sapply(0:4, function(nr){dbinom(nr, 4, 0.443)*211})
> teoreetilised
[1] 20.309687 64.611788 77.081747 40.870394  8.126384
```

Kas täishäälikute arv sõnas jaotub juhuslikult, saab kontrollida hii-ruut testiga.

```
> chisq.test(tegelikud, teoreetilised)

        Pearson's Chi-squared test

data:  tegelikud and teoreetilised
X-squared = 20, df = 16, p-value = 0.2202

Warning message:
In chisq.test(tegelikud, teoreetilised) :
  Chi-squared approximation may be incorrect
```

Andmestiku väiksuse ning tühjade lahtrite esinemise tõttu aga ei õnnestu selget vastust välja tuua. Nullhüpoteesi tõenäosus ehk p-väärtus ehk tõenäosus, et täishäälikute arv allub neljatähelistes sõnades binoomjaotusele on 22% - mis ei luba aga meil praegu midagi kummaski suunas järeldada.

Illustreerimiseks saab vähemasti tulemused tabelis ja joonisel välja kuvada.

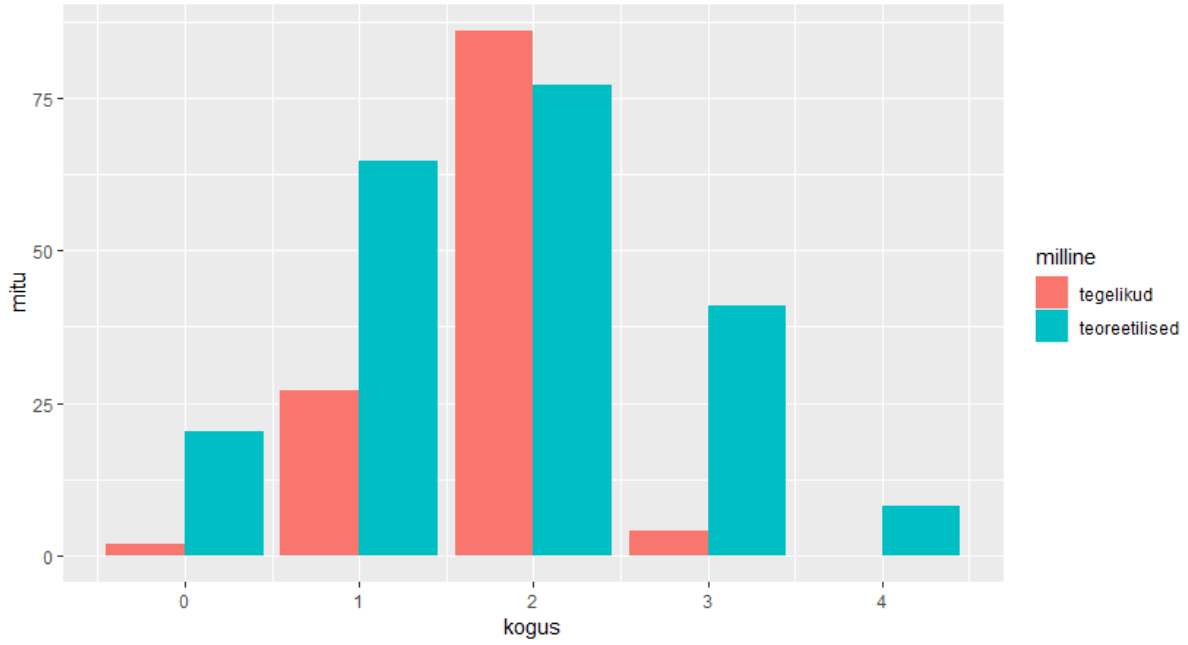
```
 tibble(kogus=0:4, tegelikud, teoreetilised)
# A tibble: 5 x 3
  kogus tegelikud teoreetilised
  <int>     <int>         <dbl>
1     0         2         20.3
2     1        27         64.6
3     2        86         77.1
4     3         4         40.9
5     4         0          8.13
```

Diagrammi tarbeks tõstame andmed pikale kujule - võtmeks "milline" ning väärtuseks "mitu", täishäälikute arvu näitav kogus jääb rea ette alles.

```
> tibble(kogus=0:4, tegelikud, teoreetilised) %>% gather(milline, mitu, -kogus)
# A tibble: 10 x 3
   kogus milline        mitu
   <int> <chr>         <dbl>
 1     0 tegelikud      2
 2     1 tegelikud     27
 3     2 tegelikud     86
 4     3 tegelikud      4
 5     4 tegelikud      0
 6     0 teoreetilised 20.3
 7     1 teoreetilised 64.6
 8     2 teoreetilised 77.1
 9     3 teoreetilised 40.9
10     4 teoreetilised  8.13
```

Sealt edasi suudab ggplot juba andmed sisse võtta ning tulpdiagrammi koostada

```
> tibble(kogus=0:4, tegelikud, teoreetilised) %>% gather(milline, mitu, -kogus) %>%
ggplot(aes(kogus, mitu, fill=milline))+geom_col(position = "dodge")
```



qbinom

Sarnaselt normaaljaotusele on ka binoomjaotuse puhul võimalik anda sisse sõnade osakaal ning küsida, et kuni mitme täishäälikuga sõnad mahuvad selle osakaalu sisse. Eelnevalt paistis, et nulli ja ühe täishäälikuga sõnad võiksid teoreetiliselt moodustada 40,2% sõnadest.

```
> pbinom(1, 4, 0.443)
[1] 0.4024714
```

Vastupidi küsides öeldakse, et 41% sõnade kokku saamiseks tuleb juba kaasata ka kahe täishäälikuga neljatähelised sõnad. 35% sõnadest tuleb ka kuni ühe täishäälikuga välja.

```
> qbinom(0.41, 4, 0.443)
[1] 2
> qbinom(0.35, 4, 0.443)
[1] 1
```

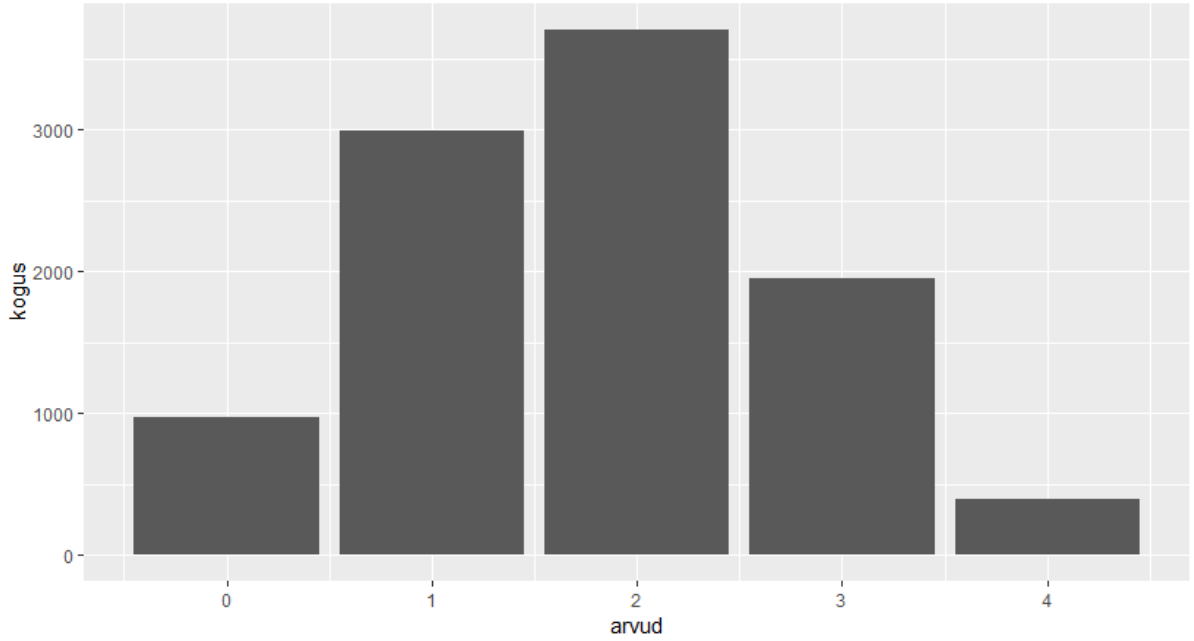
rbinom

Jaotusele vastavate juhuarvude loomiseks sobib käsklus rbinom. Parameetritena öeldakse ette mitu arvu soovitakse, seeria (siinses näites siis sõna) pikkus ning sündmuse (ehk täishääliku esinemise) tõenäosus.

```
> rbinom(10, 4, 0.443)
 [1] 2 2 1 0 1 0 3 2 1 3
```

Lisaks kümnele tuhandele jaotusele vastavale genereeritud arvule koostatud sagedusjoonis

```
> tibble(arvud=rbinom(10000, 4, 0.443)) %>% group_by(arvud) %>% summarise(kogus=n()) %>%
ggplot(aes(arvud, kogus)) + geom_col()
```



Harjutus

- Sulghäälikute osakaal sõnas on keskmiselt 0,25. Genereeri ennustatavad sulghäälikute arvud kümnes seitsmetähelises sõnas
- Leia, mitmel protsendil sõnadest võiks olla kuni kaks sulghäälikut
- Leia, mitmel protsendil sõnadest võiks olla täpselt kaks sulghäälikut
- Leia, kuni mitu sulghäälikut on ¾ sõnadest

Ühtlane jaotus

Mõnigikord on esimeseks lähendiks, et kui me ei tea, kuidas andmed on jaotunud, siis arvestame, et nad on teadaolevas vahemikus jaotunud ühtlaselt. Tüüpiliseks näiteks tuuakse sealjuures bussi ootamise aega. Siin genereerime kolmkümmend ootamise aega kestusega 0 kuni 10 minutit

```
> runif(30, 0, 10)
 [1] 6.53425563 5.93377389 6.84974829 1.95880852 9.91242426 1.01894649 7.21726539
7.28439205
 [9] 1.09905096 2.00835302 0.02995783 3.34352654 9.89536439 0.31871181 7.95551833
5.26974085
[17] 2.48233127 9.89879840 8.98220039 8.07218825 1.77151727 4.16534593 0.45706646
7.56748838
```

```
[25] 4.16563811 3.00549781 3.19464148 7.38577237 3.53665630 9.44900919
```

Küsime, millise osakaaluga on ühtlases jaotuses vahemikus 0 kuni 10 väärtused kuni 4

```
> punif(4, 0, 10)
[1] 0.4
```

Millise väärtuseni on 40% väärtustest - nagu muudegi jaotuste puhul, töötavad punif ja qunif teineteise suhtes tagurpidi

```
> qunif(0.4, 0, 10)
[1] 4
```

Ja millise väärtuseni 90% väärtustest ühtlase jaotuse juures 0 kuni 10

```
> qunif(0.9, 0, 10)
[1] 9
```

Funktsioon dunif ehk density uniform ehk ühtlase jaotuse tihedusfunktsioon näitab, kui suur osakaal väärtustest on praegu kolme ümber ühe ühiku ulatuses jaotunud. Kuna kõik väärtused on ühtlaselt jaotatud kümne ühiku peale, siis kolme juures peabki olema neid 10% ehk 0.1

```
> dunif(3, 0, 10)
[1] 0.1
```

Sama vastus ka kaheksa ümbruses

```
> dunif(8, 0, 10)
[1] 0.1
```

Kuna ühtlane jaotus praegusel juhul nulli ja kümne vahel, siis 12 juures tuleb vastuseks 0

```
> dunif(12, 0, 10)
[1] 0
```

Harjutus

- Tramm käib 7 minuti tagant. Genereeri paarkümmend võimalikku võrdtõenäolist trammi ootamise aega
- Tramm käib 7 minuti tagant, trammisõit kestab 6 minutit. Illustreeri võimalikke kohalejõudmiseks kulunud aegu.
- Tramm käib 7 minuti tagant. Trammisõit kestab keskmiselt 6 minutit, standardhälbega 1,5 minutit. Genereeri võimalikud kohale jõudmise ajad, illustreeri neid.

Poissoni jaotus

ehk jaotus, mis jäljendab juhuslike sündmuste esinemist. Mudel on paratamatult vaid lähend tegelikule olukorrale. Näiteks sulghäälikute arvu sõnas saab vaadata nii normaal- kui

poissoni jaotuse järgi - ja siis sealt pealt soovi korral edasi arvutusi teha. Normaaljaotuse puhul eeldatakse keskväärtuse juurest mõlemas suunas võrdsel moel kahanemist. Poissoni jaotuse puhul on teada, et arv alla nulli minna ei saa ning samas ülemist piiri pole otseselt määratud.

Leiame sulghäälikute arvu sõnade juures ning sulghäälikute osakaalu, saame ligikaudu 1,25 sulghäälikut sõna kohta

```
> sonad$sulghaalikuid %>% sum()
[1] 839
> sonad %>% nrow()
[1] 672
> 839/672
[1] 1.248512
```

Genereerime selle järgi 20 sõna eeldatava sulghäälikute arvu. Nagu näha, pakutakse siia sellel katsel nii nulli, ühe, kahe kui ka kolme sulghäälikuga sõnu. Uutel katsetel või suurema katsete arvu puhul võiksid sekka juhtuda ka nelja või viie või rohkemagi sulghäälikuga sõnad - aga need on lihtsalt väga haruldased

```
> rpois(20, 1.25)
 [1] 2 1 0 1 1 0 1 2 1 0 2 3 2 0 3 3 0 1 1 0
```

Küsime, et millisel osakaalul sõnadest võiks olla kuni kaks sulghäälikut. Vastuseks 87%

```
> ppois(2, 1.25)
[1] 0.8684677
```

Teine küsimus koos vastusega:

```
> #Mitmel protsendil sõnadest võiks olla 0 sulghäälikut?
> ppois(0, 1.25)
[1] 0.2865048
```

Sama tulemuse annab nulli sulghäälikuga sõnade tihendusfunktsiooni küsimine

```
> dpois(0, 1.25)
[1] 0.2865048
```

Nulli kuni ühe sulghäälikuga sõnu on 64%

```
> ppois(1, 1.25)
[1] 0.6446358
```

Ühe sulghäälikuga sõnu juhusliku jaotuse järgi 36%

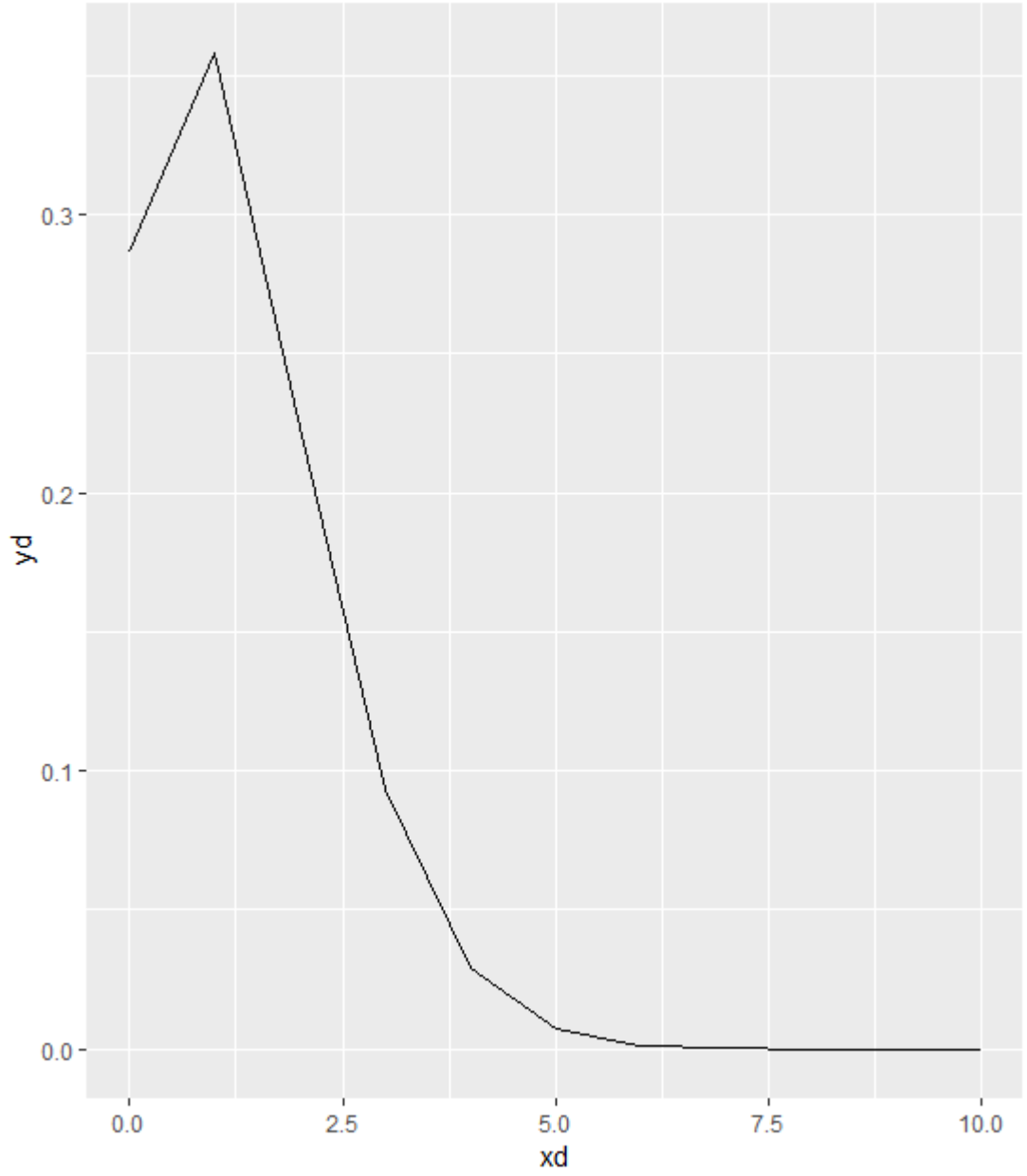
```
> dpois(1, 1.25)
[1] 0.358131
```

Sama 64% saame ka nulli ja ühe sulghäälikuga sõnade osakaalud kokku liites

```
> dpois(0, 1.25)+dpois(1, 1.25)
[1] 0.6446358
```

Poissoni funktsioon keskväärtusega (ja ühtlasi moodiga ehk tipuga) 1,25 graafikuna

```
tibble(xd=0:10) %>%
    mutate(yd=dpois(xd, 1.25)) %>%
    ggplot(aes(xd, yd))+geom_line()
```



95% sõnadest võiksid poissoni jaotuse järgi olla kuni kolme sulghäälikuga (ehk siis 5% rohkem kui 3 sulghäälikuga)

```
> qpois(0.95, 1.25)
[1] 3
```

Ning ainult ühel sõnal tuhandest võiks selle jaotuse järgi olla rohkem kui kuus sulghäälikut

```
> qpois(0.999, 1.25)
[1] 6
```

Tegelikud andmed, et kui palju millise sulghäälikute arvuga sõnu on:

```
> sonad %>% group_by(sulghaalikuid) %>% summarise(skogus=n())
# A tibble: 6 x 2
  sulghaalikuid skogus
          <dbl>  <int>
1             0    192
2             1    232
3             2    160
4             3     67
5             4     19
6             5      2
```

Tihedusfunktsiooni väärtused poissoni jaotusele tipuga 1,25

```
> sapply(0:5, function(nr){dpois(nr, 1.25)})
[1] 0.286504797 0.358130996 0.223831873 0.093263280 0.029144775 0.007286194
```

Ümberarvestus 672 sõna tarbeks, nagu neid loetelus on

```
> sapply(0:5, function(nr){dpois(nr, 1.25)}) * 672
[1] 192.531223 240.664029 150.415018  62.672924  19.585289   4.896322
```

Kumbki muutujatesse

```
> juhuslikud=sapply(0:5, function(nr){dpois(nr, 1.25)}) * 672
> shkogused=sonad %>% group_by(sulghaalikuid) %>% summarise(skogus=n()) %>% .$skogus
> juhuslikud
[1] 192.531223 240.664029 150.415018  62.672924  19.585289   4.896322
> shkogused
[1] 192 232 160  67  19   2
```

Kõrvuti pannes näeb märgatavat sarnasust. Puuduva sulghäälikuga on mõlemal juhul 192 sõna. Ühe sulghäälikuga on teoreetilises jaotuses natuke rohkem, kahe ja kolme sulghäälikuga on tegelikus loendis veidi enam, neljaseid on mõlemis jälle 19.

Praeguse 672 sõna juures ei suuda hii-ruut test üldistatavat erinevust poissoni jaotusest näidata, väidab, et 71% tõenäosusega võiksid nad sarnased olla - mida on päris palju.

```
> jsuhe=juhuslikud/sum(juhuslikud)
> jsuhe
[1] 0.287032387 0.358790484 0.224244052 0.093435022 0.029198444 0.007299611
> chisq.test(shkogused, p=jsuhe)

        Chi-squared test for given probabilities

data:  shkogused
X-squared = 2.946, df = 5, p-value = 0.7083
```

Harjutus

Folkloorieksperditsioonil leiti varasemate andmete põhjal, et ühte laulikut põhjalikult küsitledes saab tema käest üles kirjutada keskmiselt 1000 värsirida. (Millegipärast) eeldame, et võimalike üleskirjutatavate värsiridade arv võiks käia poissoni jaotuse järgi (sest alla nulli ei õnnestu kirjutada ning otsest selget ülapiiri ka ei paista)

- Genereeri võimalikud värsiridade arvud paarikümne lauliku puhul. Illustreeri graafiliselt
- Kui suur osakaal laulikukülastustest võiks anda alla 100 värsirea
- Kui suur osakaal laulikukülastustest võiks anda üle 10000 värsirea
- Mitmel protsendil külastustest on lootust üles kirjutada vähemasti 500 värsirida

(märkus - paistab, et siin näites jaotus ja tegelikkus ei lange usutavalt kokku - samas pakub võimalusi, kuidas valemit (ebatäpsuse) poole korrigeerida)

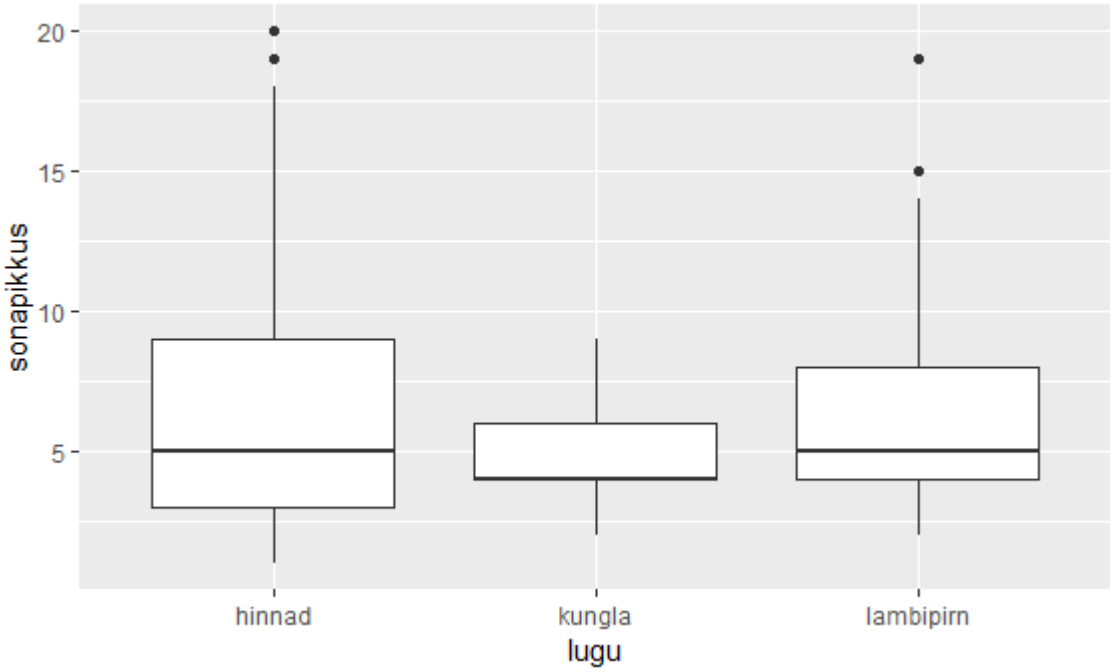
ANOVA

ANalysis of Variance ehk dispersioonanalüüs võimaldab aritmeetilisi keskmisi võrrelda rohkemate gruppide vahel. Näitena loeme sisse andmestiku, kus lisaks eelnevalt tuttavale Kungla rahva loole ning Lambipirni anekdoodile on juures majandusartikkel hindade kohta.

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikkused_haalikud.txt")
```

Sõnapikkusi illustreeriv joonis

```
> sonad %>% ggplot(aes(lugu, sonapikkus)) + geom_boxplot()
```



ning juhuslikud kümme rida andmestikust

```
> sonad %>% sample_n(10)
# A tibble: 10 x 5
   lugu      sona       sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>           <int>         <int>          <int>
 1 hinnad    40                  2             0              0
 2 lambipirn leti                4             2              1
 3 hinnad    %                   1             0              0
 4 hinnad    küte                4             2              2
 5 kungla    laulan              6             3              0
 6 lambipirn ja                  2             1              0
 7 hinnad    madal               5             2              1
 8 hinnad    veebruaris         10             5              1
 9 lambipirn ja                  2             1              0
10 lambipirn väljub              6             2              1
```

Käskluse esialgne kuju - näidatakse kogu keskmisest erinevuste ruutude summat (8866.235) ning osa, mis seotud sellega, millise looga tegemist (92.979). Silmaga vaadates vaid veidi rohkem kui sajandik.

```
> aov(sonapikkus~lugu, data=sonad)
Call:
   aov(formula = sonapikkus ~ lugu, data = sonad)

Terms:
                    lugu Residuals
Sum of Squares    92.979  8866.235
Deg. of Freedom        2       911

Residual standard error: 3.119683
Estimated effects may be unbalanced
```

Põhjalikumaks vastuseks tasub juurde kirjutada summary. Siis näeb P-väärtust, ehk nullhüpoteesi kehtimise tõenäosust - et lugu ei mõjuta sõnade pikkust. See tõenäosus on 0.00864, ehk siis rohkem kui 99% tõenäosusega võime väita, et mingi mõõdetav ja üldistatav seos sõnade pikkuse ning loo vahel on siiski olemas.

```
> summary(aov(sonapikkus~lugu, data=sonad))
             Df Sum Sq Mean Sq F value  Pr(>F)
lugu          2     93   46.49   4.777 0.00864 **
Residuals   911   8866    9.73
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Kui üldpilt käes - et mingit seost siiski tasub otsida, siis tehakse dispersioonanalüüsi puhul järel- ehk Post-Hoc testid. Üks lihtne neist TukeyHSD. Tuuakse välja lugude paarid, näidatakse erinevuse suurus, nende ülem- ja alampiir usaldusnivoo juures ning p-väärtus, ehk täenäosus, et erinevust nende kahe rühma vahel pole.

```
> TukeyHSD(aov(sonapikkus~lugu, data=sonad))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = sonapikkus ~ lugu, data = sonad)

$`lugu`
                         diff        lwr        upr     p adj
kungla-hinnad    -1.21520661 -2.1830734 -0.2473398 0.0092053
lambipirn-hinnad -0.08575938 -0.6438573  0.4723385 0.9307935
lambipirn-kungla  1.12944724  0.2322434  2.0266510 0.0089852
```

Nagu arvudelt paistab, siis lambipirni jutu ja hindade teksti vahel mõõdetav erinevus puudub, Kungla rahva laul aga teistest siiski mõõdetavalt erinev.

Harjutus

- Pange näited tööle
- Võrrelge sulghäälikute keskmist arvu sõnas eelneva faili kolme teksti vahel
- Võrrelge täishäälikute suhtelise sageduse keskmisi neis kolmes tekstis

```
> summary(aov(sulghaalikuid~lugu, data=sonad))
             Df Sum Sq Mean Sq F value   Pr(>F)
lugu          2   34.6  17.311   14.33 7.47e-07 ***
Residuals   911 1100.7   1.208
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Vahe märgatavana olemas. Lähemal uurimisel paistab, et sulghäälikute arvu koha pealt sõnas on Kungla rahva laul ja hindade artikkel pigem sarnasemad. Või õigemini - kuna Kungla rahva laul on suhteliselt lühike, siis absoluutväärtuselt isegi suurema erinevuse juures (-0.36 ja 0.27) on esimene seos vähem üldistatav

```
> TukeyHSD(aov(sulghaalikuid~lugu, data=sonad))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = sulghaalikuid ~ lugu, data = sonad)

$`lugu`
                       diff         lwr         upr     p adj
kungla-hinnad    -0.3654545 -0.70647700 -0.02443209 0.0322682
lambipirn-hinnad  0.2744785  0.07783577  0.47112114 0.0031249
lambipirn-kungla  0.6399330  0.32380825  0.95605775 0.0000070
```

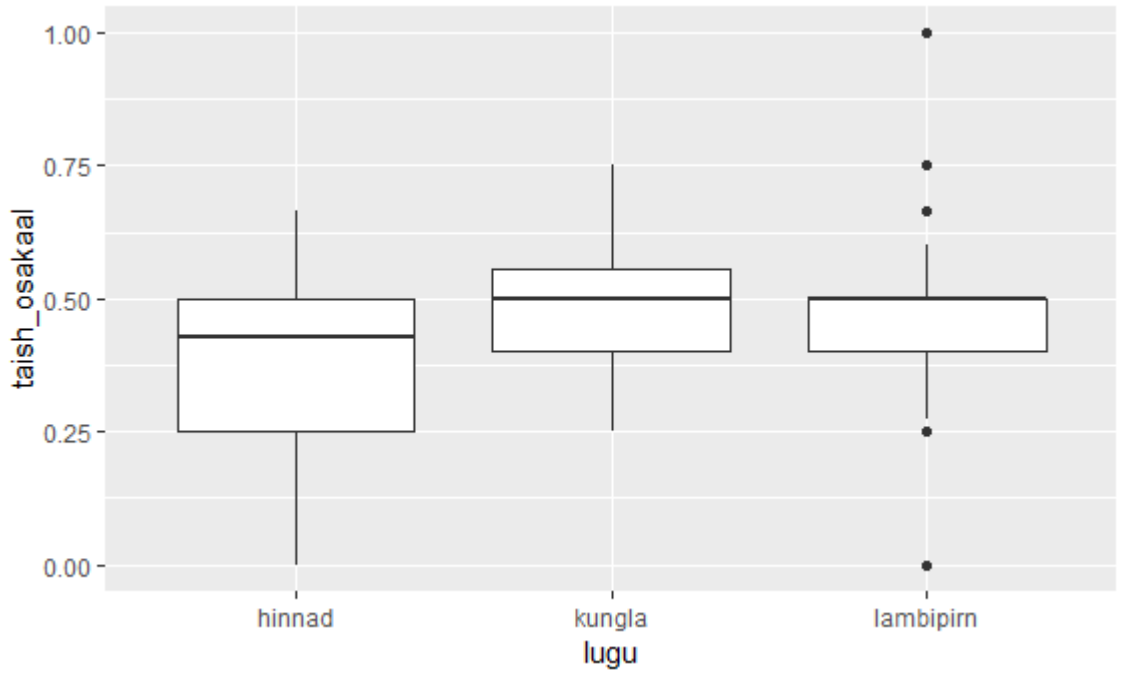
Täishäälikute osakaalu leidmiseks tuleb nende arv jagada sümbolite arvuga sõnas.

```
> summary(aov(taish_osakaal~lugu, data=sonad %>%
mutate(taish_osakaal=taishaalikuid/sonapikkus)))
             Df Sum Sq Mean Sq F value Pr(>F)
lugu          2   2.49  1.2449   48.84 <2e-16 ***
Residuals   911  23.22  0.0255
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(aov(taish_osakaal~lugu, data=sonad %>%
mutate(taish_osakaal=taishaalikuid/sonapikkus)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = taish_osakaal ~ lugu, data = sonad %>% mutate(taish_osakaal =
taishaalikuid/sonapikkus))

$`lugu`
                        diff         lwr        upr     p adj
kungla-hinnad     0.12950084  0.07996681 0.17903488 0.0000000
lambipirn-hinnad  0.11658939  0.08802674 0.14515205 0.0000000
lambipirn-kungla -0.01291145 -0.05882906 0.03300615 0.7866688
```

Sedakorda paistab Kungla rahva laul olema suhteliselt sarnane lambipirni jutule.

Joonis täishäälikute osakaalu jaotuse iseloomustamiseks vastavalt loole

```
> sonad %>% mutate(taish_osakaal=taishaalikuid/sonapikkus) %>% ggplot(aes(lugu,
taish_osakaal)) + geom_boxplot()
```



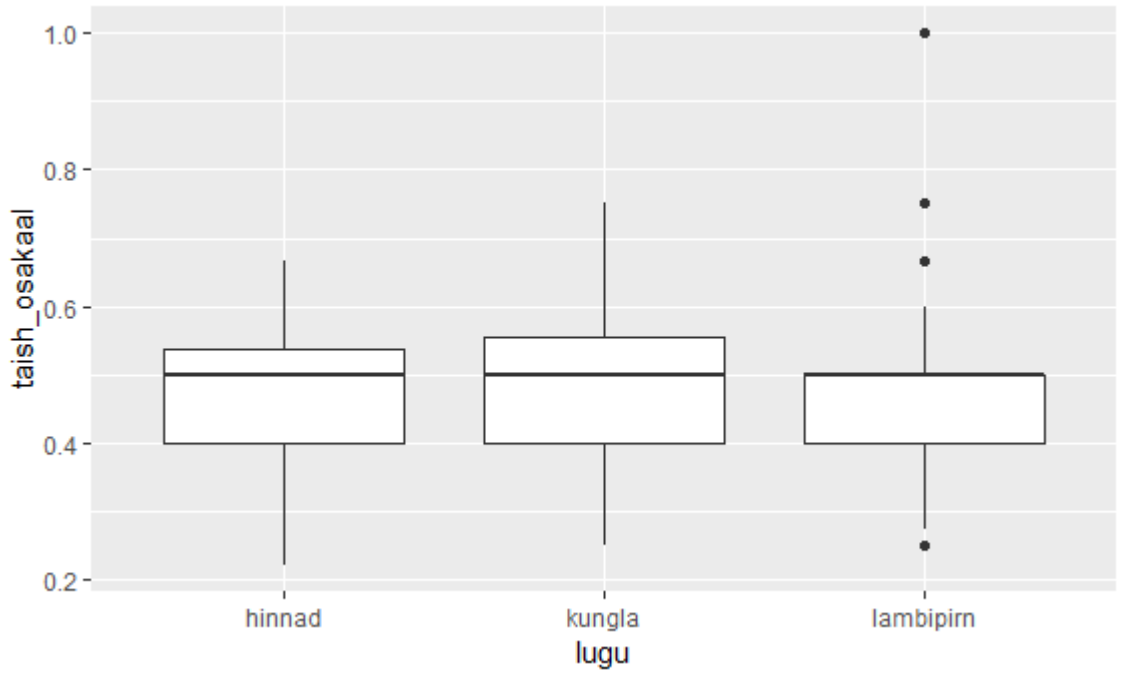
Paistab välja, et Kungla rahva juures täishäälikute osakaal väga madalale ei lähe, teiste puhul esineb ka sõnu, kus pole ühtegi täishäälikut. Kuna eesti keele puhul sellised sõnad tunduvad haruldased, siis uurime lähemalt, et millega tegu

```
> sonad %>% filter(taishaalikuid==0, sulghaalikuid==0)
# A tibble: 64 x 5
   lugu      sona  sonapikkus taishaalikuid sulghaalikuid
   <chr>     <chr>      <int>         <int>         <int>
 1 lambipirn --             2             0             0
 2 lambipirn ).             2             0             0
 3 lambipirn 30-40          5             0             0
 4 lambipirn ??             2             0             0
 5 lambipirn --             2             0             0
 6 lambipirn !”             2             0             0
 7 hinnad    3.4            3             0             0
 8 hinnad    2              1             0             0
 9 hinnad    2016.          5             0             0
10 hinnad    2019.          5             0             0
# ... with 54 more rows
```

Nagu näha, siis tegemist arvude ja igasugu muude sümbolikombinatsioonidega. "Pärissõnade" võrdlemiseks jätame sisse ainult sellised, kus täishäälikud ka olemas

```
> sonad %>% filter(taishaalikuid>0) %>% mutate(taish_osakaal=taishaalikuid/sonapikkus) %>%
ggplot(aes(lugu, taish_osakaal)) + geom_boxplot()
```

Pilt tuli juba märgatavalt ühtlasem



Sama lugu arvulise hinnangu kohta. P väärtusega 47% ei luba enam mingit erinevust üldistada

```
> summary(aov(taish_osakaal~lugu, data=sonad %>% filter(taishaalikuid>0) %>%
mutate(taish_osakaal=taishaalikuid/sonapikkus)))
             Df Sum Sq Mean Sq F value Pr(>F)
lugu          2  0.022 0.01096    0.75  0.473
Residuals   847 12.385 0.01462
```

Samuti ei tule üldistavat erinevust välja tekstipaaride vahel

```
> TukeyHSD(aov(taish_osakaal~lugu, data=sonad %>% filter(taishaalikuid>0) %>%
mutate(taish_osakaal=taishaalikuid/sonapikkus)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = taish_osakaal ~ lugu, data = sonad %>% filter(taishaalikuid > 0) %>%
mutate(taish_osakaal = taishaalikuid/sonapikkus))

$`lugu`
                         diff         lwr        upr     p adj
kungla-hinnad     0.018248552 -0.02064482 0.05714192 0.5133368
lambipirn-hinnad  0.010103880 -0.01386311 0.03407087 0.5835191
lambipirn-kungla -0.008144672 -0.04294457 0.02665523 0.8467653
```

Sealtkaudu saame järeldada, et majandustekste täishäälikusisalduse järgi eristada ei saa, küll aga võiks see õnnestuda võrreldes näiteks uurides arvude osakaalu tekstis.

Tulpdiagramm keskmiste, standardhälvete ja standardvigadega

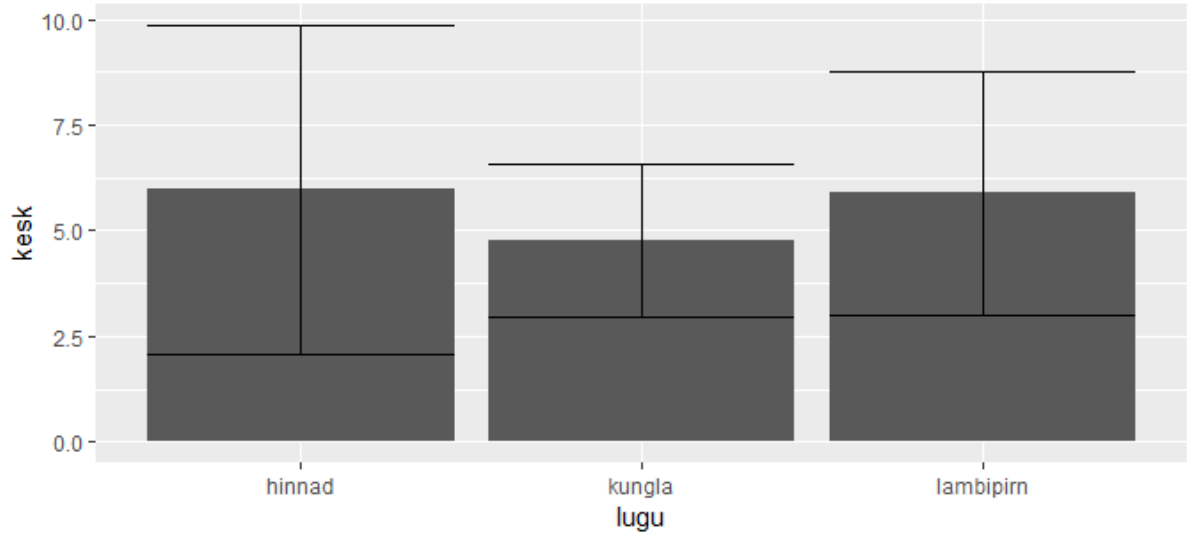
Karpdiagramm ja histogramm annavad andmete jaotumisest ilusa pildi, ANOVA arvutuskäik käib aga keskväärtuse ja standardhälvete kaudu. Arvutame need illustreerimiseks välja.

```
keskmised <- sonad %>% group_by(lugu) %>% summarise(kesk=mean(sonapikkus),
sh=sd(sonapikkus))
```

Tulemus:

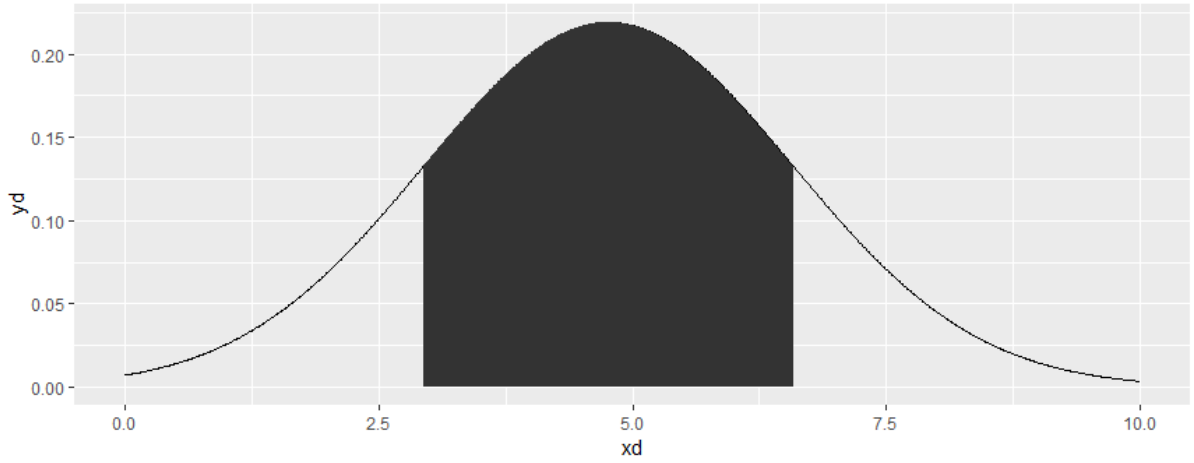
```
keskmised
# A tibble: 3 x 3
  lugu       kesk    sh
  <chr>     <dbl> <dbl>
1 hinnad     5.98  3.90
2 kungla     4.76  1.82
3 lambipirn  5.89  2.88
```

```
keskmised %>% ggplot(aes(lugu, kesk)) + geom_col() +
    geom_errorbar(aes(ymin=kesk-sh, ymax=kesk+sh))
```



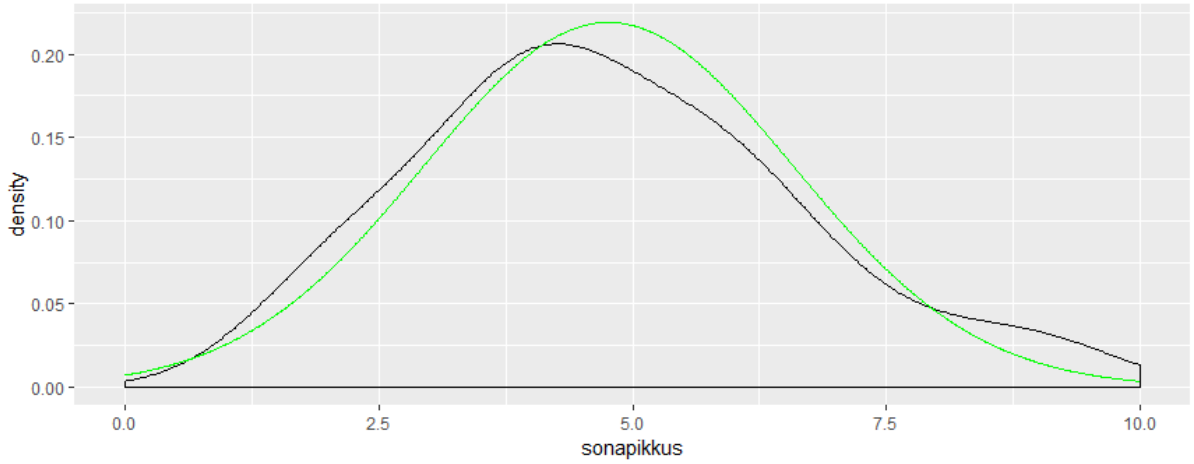
Siin siis näha, kus sõnad jaotuse järgi paikneksid. Normaaljaotusele lähenemise korral on aritmeetilise keskmise +/- ühe standardhälbe kaugusel ligikaudu ⅔ väärtustest.

```
tibble(xd=seq(0, 10, length=1000)) %>%
  mutate(yd=dnorm(xd, mean=4.76, sd=1.82)) %>%
  ggplot(aes(xd, yd))+
   geom_line() + geom_area(aes(y=ifelse((xd>4.76-1.82) & (xd<4.76+1.82), yd, 0)))
```



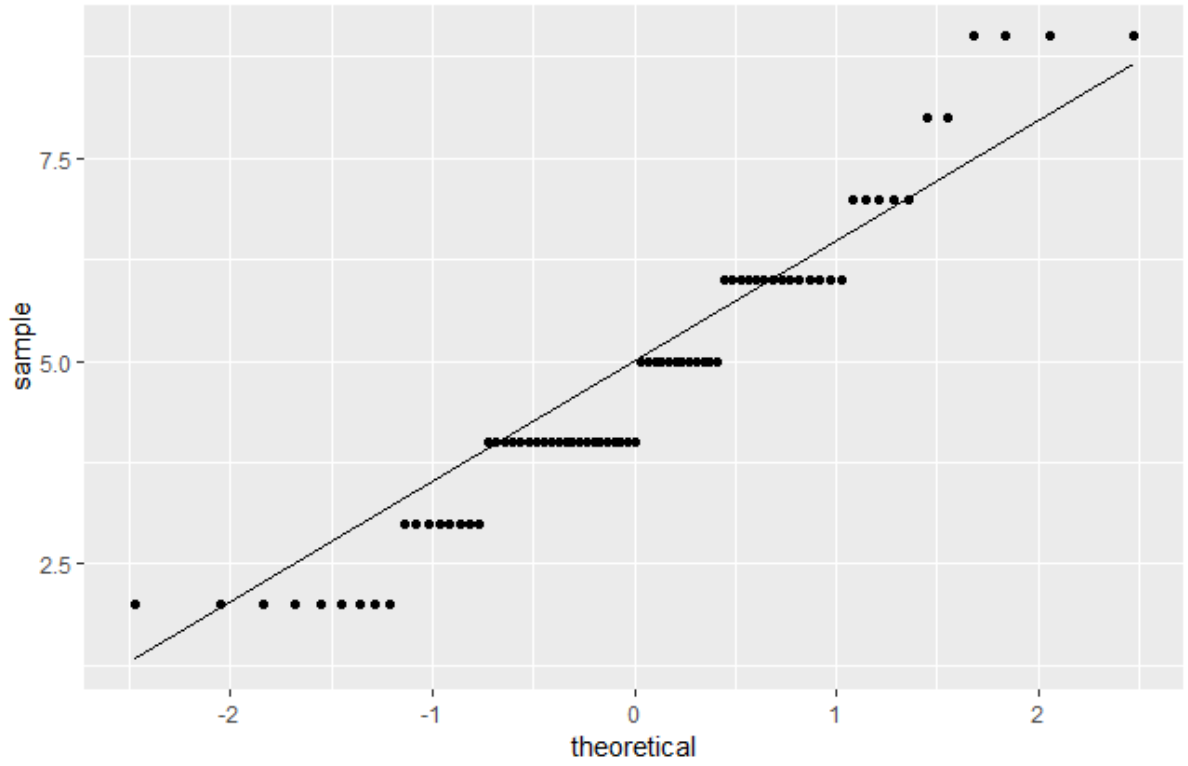
Võrdlusena ka Kungla rahva sõnapikkuste jaotuse (must) ning sama aritmeetilise keskmise ning standardhälbega normaaljaotuse (roheline) joonis

```
sonad %>% filter(lugu=="kungla") %>%
  ggplot((aes(sonapikkus))) + geom_density(adjust=1.5) + xlim(0, 10) +
     geom_line(aes(xd, yd), data=tibble(xd=seq(0, 10, length=1000)) %>%
      mutate(yd=dnorm(xd, mean=4.76, sd=1.82)), color="green")
```



Sama võrdlus kvantiil-kvantiil joonise kaudu, mis vaikimisi võrdleb normaaljaotusega. Pilt teise kujuga, aga sama sisuga. Kahetähelisi sõnu on normaaljaotumisega võrreldes rohkem, keskmise pikkusega sõnade ajaks vastab tegelik jaotus enamvähem normaaljaotusele. 7- ja 8-täheliste sõnade vähesus Kungla rahva loos sikutab kvantiilivõrdluse teisele poole ühtlase jaotuse joont ning üheksatähelised toovad tasakaalu jälle tagasi

```
sonad %>% filter(lugu=="kungla") %>%
ggplot(aes(sample=sonapikkus))+geom_qq()+geom_qq_line()
```

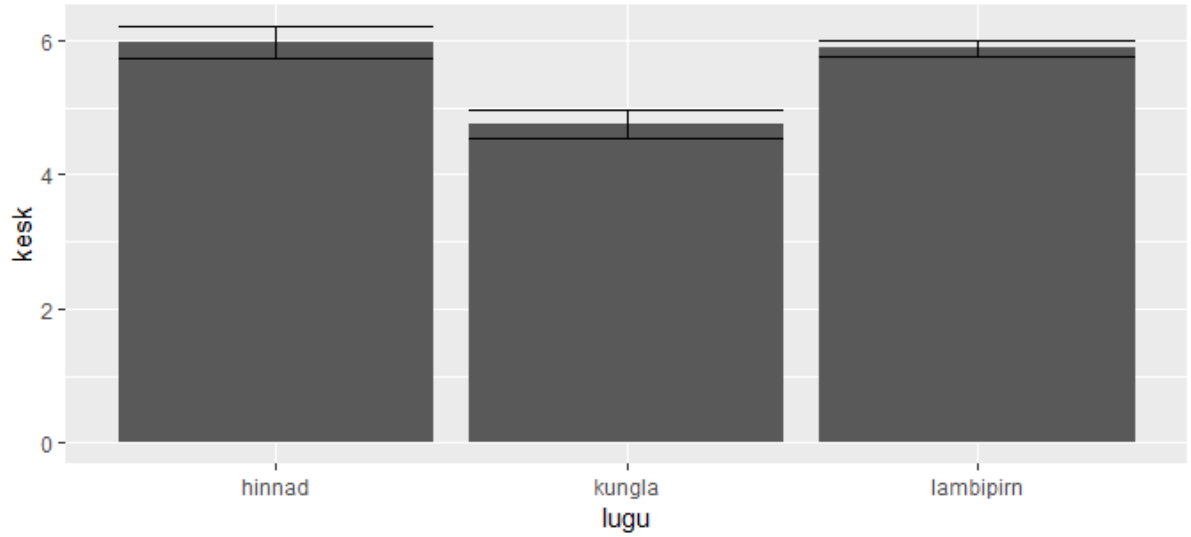


Arvutus koos standardveaga - et kuivõrd eri valimite puhul võiks aritmeetiline keskmine kõikuda

```
keskmised <- sonad %>% group_by(lugu) %>% summarise(kesk=mean(sonapikkus),
sh=sd(sonapikkus), kogus=n(), standardviga=sh/sqrt(kogus))
keskmised
# A tibble: 3 x 5
  lugu       kesk    sh kogus standardviga
  <chr>     <dbl> <dbl> <int>        <dbl>
1 hinnad     5.98  3.90   242        0.251
2 kungla     4.76  1.82    75        0.210
3 lambipirn  5.89  2.88   597        0.118
```

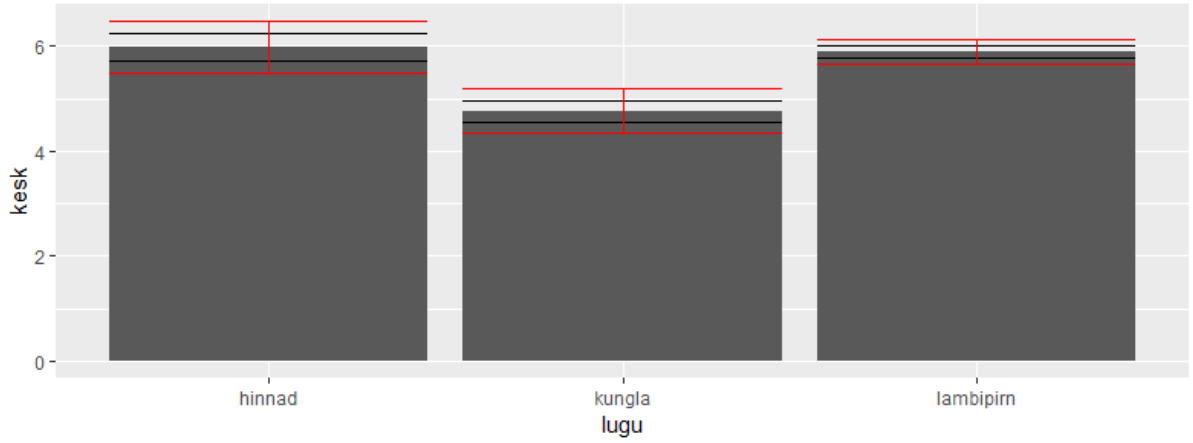
Andmete põhjal joonis

```
> keskmised %>% ggplot(aes(lugu, kesk)) + geom_col() +
geom_errorbar(aes(ymin=kesk-standardviga, ymax=kesk+standardviga))
```



Juurde ka joonis kahekordse standardveaga - et kuhu võiks keskmine sattuda 95% juhtudest

```
> keskmised %>% ggplot(aes(lugu, kesk)) + geom_col() +
geom_errorbar(aes(ymin=kesk-standardviga, ymax=kesk+standardviga)) +
geom_errorbar(aes(ymin=kesk-2*standardviga, ymax=kesk+2*standardviga), color="red")
```



Harjutus

- Tehke näited läbi
- Arvutage standardhälve ja standardviga sulghäälikute sõna keskmise arvu kohta igas tekstis.
- Kuvage sulghäälikute arvu keskväärtus ja standardviga joonisele lugude kaupa
- Kuvage joonis ka kahekordse standardveaga

Tabelite ühendamine, keelekorpuse andmed

Vähegi suuremas süsteemis on andmed sageli mööda tabeleid laiali. Näide õppijakeele korpuse tekstide kohta. Ühes failis tekstide metaandmed

```
dokmeta=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt")
```

teises sõnaliikide sagedused tekstide kaupa

```
doksonaliigid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt")
```

Failialgused tutvumiseks:

```
> head(dokmeta)
# A tibble: 6 x 13
  kood                korpus    tekstikeel tekstityyp elukoht taust vanus  sugu  emakeel kodukeel keeletase haridus abivahendid
  <chr>               <chr>     <chr>      <chr>      <chr>   <chr> <chr>  <chr> <chr>   <chr>    <chr>     <chr>   <chr>
1 doc_100636852915_item cFOoRQekA eesti      essee      idaviru op    kuni18 naine vene    vene     B         pohi    ei
2 doc_100636852916_item cFOoRQekA eesti      muu        idaviru op    kuni18 naine vene    vene     B         pohi    ei
3 doc_100636852917_item cFOoRQekA eesti      essee      idaviru op    kuni18 naine vene    vene     B         pohi    ei
4 doc_1010138197_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     A         kesk    ei
5 doc_1010138198_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     B         kesk    ei
6 doc_1010138199_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     A         kesk    ei
```

Täht veeru pealkirjana tähistab sõnaliiki

```
> head(doksonaliigid)
# A tibble: 6 x 18
  kood                    A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z kokku
  <chr>               <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 doc_100636852915_item    25     0    14     0     3     0    19     5     3    17    54     0    35     0     0    36   211
2 doc_100636852916_item     4     0     5     0     4     0    12     1     3    14    31     0    22     0     0    21   117
3 doc_100636852917_item     9     0     6     0     2     0    13     1     3    17    53     0    25     0     2    27   158
4 doc_1010138197_item      46     7    50     4    20     0    38     3     2    34   183     0   126     0     2   184   699
5 doc_1010138198_item      43     7    49     4    21     0    37     6     2    39   182     0   129     0     2   177   698
6 doc_1010138199_item      45     7    51     4    20     0    38     4     2    37   180     1   132     0     2   185   708
```

Tuntumad neist V - verb/tegusõnad ja S - substantiiv/nimisõna

Ühendame kaks tabelit mõlemas tabelis esineva tulba "kood" järgi, küsime järgemööda välja iga teksti tüübi ning tegusõnade arvu

```
koos=dokmeta %>% inner_join(doksonaliigid, by="kood")
koos %>% select(tekstityyp, V)
# A tibble: 12,724 x 2
   tekstityyp     V
   <chr>      <int>
 1 essee         35
 2 muu           22
 3 essee         25
 4 muu          126
 5 muu          129
 6 muu          132
 7 muu          125
 8 referaat     791
 9 NA             0
10 essee         92
```

Kuna tekstid on eri pikkustega, siis võrreldavad on pigem tegusõnade osakaalud ehk iga teksti juures nende suhe teksti sõnade üldarvu. Käsu na.omit() abil eemaldame puuduvate väärtustega read tabelist.

```
> koos %>% mutate(tegusonasuhe=V/kokku) %>% select(tekstityyp, tegusonasuhe) %>% na.omit()
# A tibble: 4,349 x 2
   tekstityyp tegusonasuhe
   <chr>              <dbl>
 1 essee              0.166
 2 muu                0.188
 3 essee              0.158
 4 muu                0.180
 5 muu                0.185
 6 muu                0.186
 7 muu                0.181
```

```
 8 referaat          0.108
 9 essee             0.246
10 essee             0.204
```

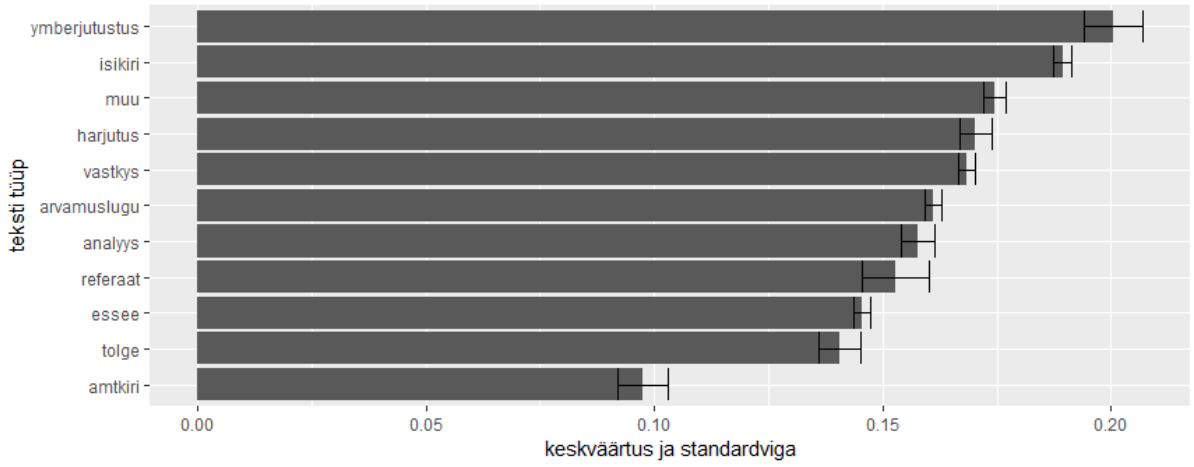
Harjutus

- Illustreerige tegusõnade suhtarvu sõltuvust tekstitüübist, kuvage karpdiagramm ning näidake erinevusi ANOVA abil.
- Tehke sama nimisõnade (S) puhul. Võrrelge lisaks nimisõnade sageduse sõltuvust teksti autori emakeelest. Jätke uuringusse ainult eestikeelsed tekstid.
- Vaadake, milliseid järeldusi saab tulemustest teha. Uurige võimalikke anomaaliaid ja nende põhjusi ning püüdke need üldpildist eraldada.

Andmestiku illustreerimiseks tegusõnade suhteliste sageduste aritmeetilised keskmised ning standardvead (standardhälve jagatud ruutjuurega elementide arvust). Hulk käske üksteise järel reas, aga nii võimalik ühe korraga tulemuseni jõuda.

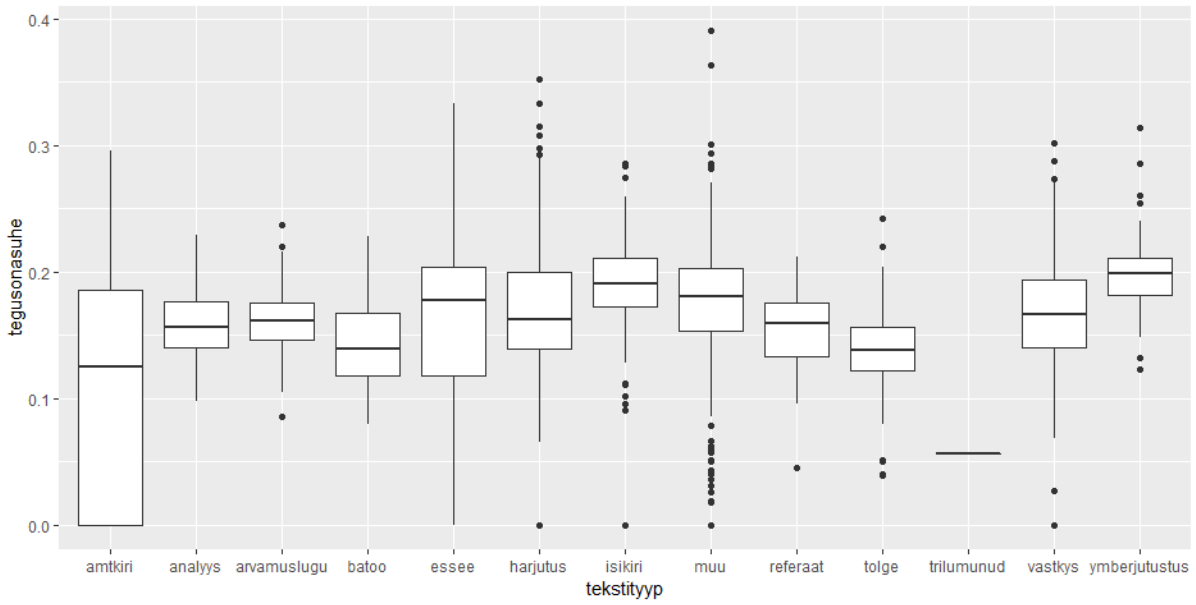
Tekstitüüpide järjestamiseks aritmeetiliste keskväärtuste järgi tuli `ggplot`-i aes-i juures `x` uue faktorina arvutada ja järjestuseks määrata tekstitüübid keskväärtuste järjekorras. Et selle peale jääksid sildid mõistlikuks, tuleb need `xlab` ja `ylab`-käsklustega üle kirjutada.

```
koos %>% mutate(tegusonasuhe=V/kokku) %>% select(tekstityyp, tegusonasuhe) %>%
  na.omit() %>% group_by(tekstityyp) %>%
  summarise(kesk=mean(tegusonasuhe), sh=sd(tegusonasuhe), kogus=n()) %>%
  filter(kogus>10) %>%
  ggplot(aes(x=factor(tekstityyp, levels=tekstityyp[order(kesk)]), y=kesk)) + geom_col() +
    geom_errorbar(aes(ymin=kesk-sh/sqrt(kogus), ymax=kesk+sh/sqrt(kogus))) +
    xlab("teksti tüüp") + ylab("keskväärtus ja standardviga") + coord_flip()
```



Karpdiagramm ka

```
koos %>% mutate(tegusonasuhe=V/kokku) %>% select(tekstityyp, tegusonasuhe) %>% na.omit()
%>% ggplot(aes(tekstityyp, tegusonasuhe)) + geom_boxplot()
```




```
> tsuhted <- koos %>% mutate(tegusonasuhe=V/kokku) %>% select(tekstityyp, tegusonasuhe) %>%
na.omit()
> summary(aov(tegusonasuhe~tekstityyp, data=tsuhted))
              Df Sum Sq Mean Sq F value Pr(>F)
tekstityyp    12  1.991 0.16593   32.49 <2e-16 ***
Residuals   4336 22.143 0.00511
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(aov(tegusonasuhe~tekstityyp, data=tsuhted))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = tegusonasuhe ~ tekstityyp, data = tsuhted)

$`tekstityyp`
                                    diff          lwr          upr     p adj
analyys-amtkiri             0.0601081607  0.025645394  0.094570927 0.0000007
arvamuslugu-amtkiri         0.0635517880  0.041362265  0.085741311 0.0000000
batoo-amtkiri               0.0487173942 -0.070498278  0.167933066 0.9805056
essee-amtkiri               0.0479972563  0.033433637  0.062560876 0.0000000
harjutus-amtkiri            0.0727755663  0.052454788  0.093096344 0.0000000
isikiri-amtkiri             0.0917756783  0.072758193  0.110793164 0.0000000
muu-amtkiri                 0.0769767977  0.059536884  0.094416711 0.0000000
referaat-amtkiri            0.0552202886  0.007640917  0.102799660 0.0078968
tolge-amtkiri               0.0430116374  0.012118868  0.073904406 0.0002982
trilumunud-amtkiri         -0.0408892578 -0.278148942  0.196370427 0.9999969
```

Arvutustest paistab välja, et seos teksti tüübiga on selge. Samas mõnede tüüpide, eriti ametlike kirjade juures on märgatav osa tekste sootuks tegusõnadeta - mis torkab silma.

```
> koos %>% filter(tekstityyp=="amtkiri") %>% .$V
  [1]  81  85 101  66  83  75  83  92  99  86  64  86 114  85  92  19   9  72   0   0   0   0   0   0
 [25]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  96   3  16  10  16  35  32
 [49]  30  31  32  31  22  22  22  35  20  20  27  28  42  29  25  14  26  23  37  37  39  36  34  25
 [73]  27  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 [97]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  35   8  25  21
[121]  20  24  23  28  16  15  21  11  16  23  27  14  31  33  45  30  41  30  32  39  36  30  40  41
[145]  45  14  18  19  12  10  20  26  26  39   5  24  11  14  21  20  37  28   5  12  20  40  67  72
```

```
[169]  47  72  95  40  53  54  55  62  64  28  25  25  11  11  24  15  18   3   0   0   0   0   0   0
[193]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
[217]   0   0   0   0   0   0  29  27   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
[241]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
[265]   0   0   7   8  16  21  74  25  13  10  26  10  12  27  26  33  32  33  25  22  30  17  30  23
[289]  22  44  32  35  32  37  26  30  33  24  36  21  27  23   5   7
```

Teksti pikkused samas neil täiesti märgatavad

```
> koos %>% filter(tekstityyp=="amtkiri", V==0) %>% .$kokku
  [1] 201 244 265 243 282 274 239 211 241 278 234 255 233 250 238 237 242 253 229 291 254 226 233 240
 [25] 238 236 243 259 230 240 222 288 254 292 276 287 271 239 266 268 257 241   0 229 213 240 247 254
 [49] 227 303 235 239 223 233 260 214 177 230 243 224 259 258 195 314 252   0 234 235 188 232 258 230
 [73] 219 194 231 237 226 249 228 216 221 195 210 235 209 226 246 261 208 275 277 262 190 254 214 246
 [97] 260 300 247 227 244 234 234 235 235 253 266 236 231 244 272 268 187 240 212 239 217 255 204 249
[121] 209 263 228 198 248 237 214 269 270 241 217 231 272 239 228 245 253 255 186 218 239 240 298
```

Küsime vastavate tekstide koodid

```
> koos %>% filter(tekstityyp=="amtkiri", V==0) %>% .$kood
  [1] "doc_18538799003_item"  "doc_18538799005_item"  "doc_18538799007_item"  "doc_18538799009_item"
  [5] "doc_18538799016_item"  "doc_18538799019_item"  "doc_18538799020_item"  "doc_18538799021_item"
```

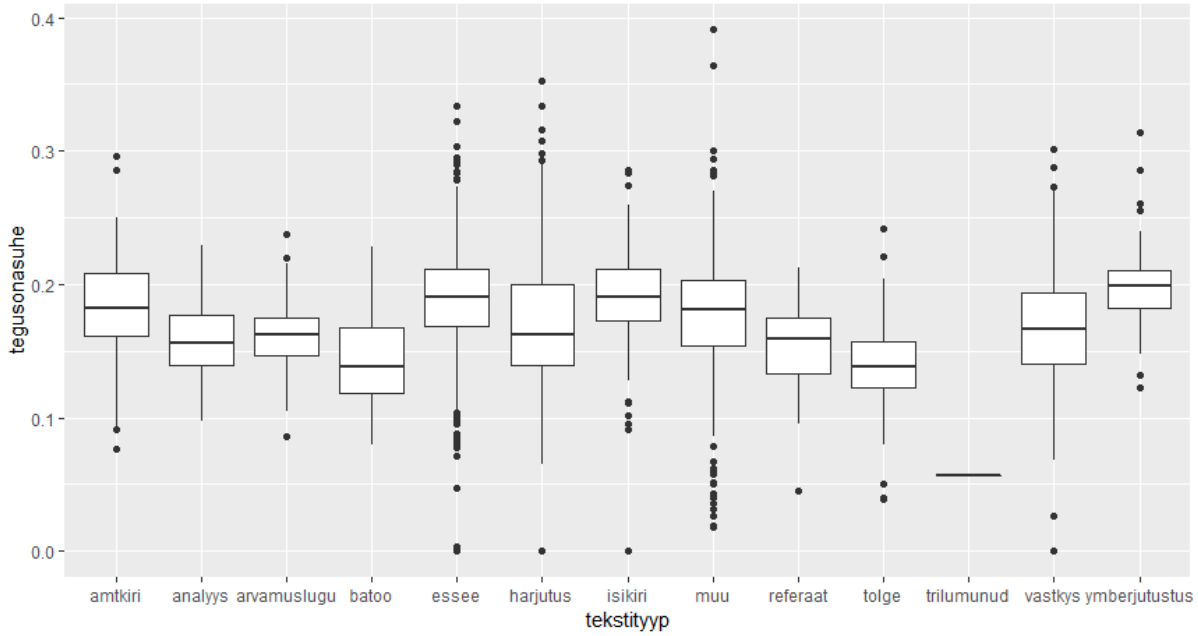
Ning vaatame neist esimese sisu avalikus veebis

```
http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_18538799003_item.txt
```

Задание 1. (120 слов) Вы учавствуете в организации международной молодёжной конференции.Напишите ответ на письмо Ирины Петровой из Пскова, в котором она просит вас дать информацию о теме конференции, о месте и времени её проведения, о сроках регистрации для участия в конференции, об участниках (какого они возраста, откуда родом, каковы их интересы), о возможностях проживания и питания во время конференции, о культурной программе, предлагаемой участникам.

Selgub, et tegemist venekeelse tekstiga, mille kohta eesti analüsaator polegi suutnud tegusõnu määrata. Järelikult tuleb eelneva päringu juures täpsustada, et soovime uurida vaid eestikeelseid tekste. Pilt muutus mõnevõrra.

```
koos %>% filter(tekstikeel=="eesti") %>% mutate(tegusonasuhe=V/kokku) %>%
select(tekstityyp, tegusonasuhe) %>% na.omit() %>% ggplot(aes(tekstityyp, tegusonasuhe)) +
geom_boxplot()
```



Edasi paistab, et leidub ka eestikeelseid tekste, milles pole ühtegi verbi

```
> koos %>% filter(tekstikeel=="eesti", V==0) %>% na.omit() %>% select(kood, tekstityyp, V,
kokku)
# A tibble: 13 x 4
   kood                  tekstityyp     V kokku
   <chr>                 <chr>      <int> <int>
 1 doc_248823787592_item isikiri        0     4
 2 doc_491521501919_item harjutus       0   188
 3 doc_540371162164_item referaat       0     0
 4 doc_542009824765_item harjutus       0    23
 5 doc_542009824766_item harjutus       0    30
 6 doc_542009824767_item harjutus       0    99
```

Vaatame ühele sellisele tekstile otsa. Nimekirjas teine tekst, kus peaks olema 188 sõna ja mitte ükski neist tegusõna

```
http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_491521501919_item.txt
```

```
Onu - onusid  Tädi - tädisid  Sõber - sõbru  Laps - lapsi  Naine - naisi  Mees - mehi
Vanur - vanur  Aasta - aastaid  Päev - päevi  Kuu - kuuid  Tund - tunde  Minut - minuteid
Öö - ööid  Sekund - sekundeid  Auto - autosid  Tramm - tramme  Buss - busse  Rong - ronge
Lennuk - lennukeid  Takso - taksosid  Takso - taksosid
```

Paistab, et tegemist on harjutusega, kus kirjas nimisõnade ainsused ja mitmused - järelikult peabki tekst selline olema.

Vaatame, ANOVA nüüd tekstitüüpide tegusõnade keskmise sisalduse erinevuste kohta ütleb:

```
summary(aov(tegusonasuhe~tekstityyp, data=koos %>% filter(tekstikeel=="eesti" ) %>%
na.omit() %>% mutate(tegusonasuhe=V/kokku)))
              Df Sum Sq Mean Sq F value Pr(>F)
```

```
tekstityyp    10  0.562 0.05617   35.11 <2e-16 ***
Residuals   2813  4.500 0.00160
```

Endiselt erinevad tekstid rühmiti märgatavalt.

Lähem uuring Tukey Honestly Siginficant Difference abil:

```
> TukeyHSD(aov(tegusonasuhe~tekstityyp, data=koos %>% filter(tekstikeel=="eesti" ) %>%
na.omit() %>% mutate(tegusonasuhe=V/kokku)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = tegusonasuhe ~ tekstityyp, data = koos %>% filter(tekstikeel == "eesti")
%>% na.omit() %>% mutate(tegusonasuhe = V/kokku))

$`tekstityyp`
                              diff           lwr          upr     p adj
analyys-amtkiri        -0.025864656 -0.0462071677 -0.005522143 0.0021371
batoo-amtkiri          -0.036465738 -0.1017069184  0.028775443 0.7801093
essee-amtkiri           0.010842206 -0.0001855665  0.021869978 0.0588374
harjutus-amtkiri       -0.017974975 -0.0321240252 -0.003825924 0.0021674
isikiri-amtkiri         0.007289360 -0.0054653060  0.020044025 0.7554230
muu-amtkiri            -0.006762866 -0.0188050811  0.005279350 0.7749848
referaat-amtkiri       -0.028809467 -0.0561018114 -0.001517123 0.0283804
tolge-amtkiri          -0.050779409 -0.0699049250 -0.031653892 0.0000000
```

Tulemused ilusasti olemas, aga neid mugavaks vaatamiseks palju ja nad segamini. Paneme testi vastuse omaette muutujasse, nii saab seda vaikselt edasi uurida.

```
testitulemus=TukeyHSD(aov(tegusonasuhe~tekstityyp, data=koos %>% filter(tekstikeel=="eesti"
) %>% na.omit() %>% mutate(tegusonasuhe=V/kokku)))

> head(testitulemus$tekstityyp)
                         diff           lwr          upr       p adj
analyys-amtkiri  -0.025864656 -0.0462071677 -0.005522143 0.002137097
batoo-amtkiri    -0.036465738 -0.1017069184  0.028775443 0.780109306
essee-amtkiri     0.010842206 -0.0001855665  0.021869978 0.058837410
harjutus-amtkiri -0.017974975 -0.0321240252 -0.003825924 0.002167370
isikiri-amtkiri   0.007289360 -0.0054653060  0.020044025 0.755423019
muu-amtkiri      -0.006762866 -0.0188050811  0.005279350 0.774984799
```

Andmestik tehniliselt maatriksiks tibble-ks, indeks omaette tulbaks

```
> tabel=testitulemus$tekstityyp %>% as_tibble() %>%
add_column(paar=rownames(testitulemus$tekstityyp))
> head(tabel)
# A tibble: 6 x 5
      diff       lwr      upr `p adj` paar
     <dbl>     <dbl>    <dbl>   <dbl> <chr>
1 -0.0259  -0.0462   -0.00552 0.00214 analyys-amtkiri
2 -0.0365  -0.102     0.0288  0.780   batoo-amtkiri
3  0.0108  -0.000186  0.0219  0.0588  essee-amtkiri
4 -0.0180  -0.0321   -0.00383 0.00217 harjutus-amtkiri
5  0.00729 -0.00547   0.0200  0.755   isikiri-amtkiri
6 -0.00676 -0.0188    0.00528 0.775   muu-amtkiri
```

Soovides rohkem ridu korraga näha, võib print-käsule vastava n-parameetri anda

Andmed järjestatud p-adj ehk olulisuse järgi. Ülakomad ümber, kuna muidu tühikuga tulbanimi ei toimi

```
> print(tabel %>% arrange(`p adj`), n=30)
# A tibble: 55 x 5
      diff       lwr      upr  `p adj` paar
     <dbl>     <dbl>    <dbl>    <dbl> <chr>
 1 -0.0508 -0.0699   -0.0317  0.       tolge-amtkiri
 2 -0.0288 -0.0393   -0.0184  0.       harjutus-essee
 3 -0.0176 -0.0249   -0.0103  0.       muu-essee
 4 -0.0616 -0.0782   -0.0451  0.       tolge-essee
 5 -0.0257 -0.0327   -0.0187  0.       vastkys-essee
 6 -0.0581 -0.0758   -0.0403  0.       tolge-isikiri
 7 -0.0221 -0.0316   -0.0126  0.       vastkys-isikiri
 8 -0.0440 -0.0613   -0.0268  0.       tolge-muu
 9  0.0359  0.0188    0.0531  0.       vastkys-tolge
10  0.0686  0.0418    0.0954  0.       ymberjutustus-tolge
11  0.0253  0.0130    0.0375  7.07e-10 isikiri-harjutus
12  0.0367  0.0187    0.0547  1.70e- 9 essee-analyys
13 -0.0328 -0.0516   -0.0140  1.12e- 6 tolge-harjutus
14  0.0332  0.0141    0.0522  1.28e- 6 isikiri-analyys
15  0.0437  0.0160    0.0714  2.21e- 5 ymberjutustus-analyys
16 -0.0397 -0.0652   -0.0141  3.33e- 5 referaat-essee
17  0.0358  0.0122    0.0594  5.55e- 5 ymberjutustus-harjutus
18  0.0326  0.0104    0.0549  1.26e- 4 ymberjutustus-vastkys
19 -0.0141 -0.0238   -0.00431 1.85e- 4 muu-isikiri
20  0.0466  0.0135    0.0798  3.20e- 4 ymberjutustus-referaat
21 -0.0361 -0.0625   -0.00974 5.47e- 4 referaat-isikiri
22 -0.0259 -0.0462   -0.00552 2.14e- 3 analyys-amtkiri
23 -0.0180 -0.0321   -0.00383 2.17e- 3 harjutus-amtkiri
24 -0.0148 -0.0267   -0.00299 2.76e- 3 vastkys-amtkiri
25  0.0246  0.00223   0.0469  1.76e- 2 ymberjutustus-muu
26 -0.0288 -0.0561   -0.00152 2.84e- 2 referaat-amtkiri
27 -0.0249 -0.0487   -0.00111 3.12e- 2 tolge-analyys
28  0.0191  0.000500  0.0377  3.82e- 2 muu-analyys
29  0.0108 -0.000186  0.0219  5.88e- 2 essee-amtkiri
30  0.0112 -0.000294  0.0227  6.39e- 2 muu-harjutus
```

Mugavaks vaatamiseks leitakse kõigepealt tegusõnade osakaalud

```
tsuhted=koos %>% filter(tekstikeel=="eesti") %>%
   mutate(tegusonasuhe=V/kokku) %>% select(tekstityyp, tegusonasuhe) %>% na.omit()

head(tsuhted)
# A tibble: 6 x 3
  tekstityyp tegusonasuhe jarjestatud_tyyp
  <chr>             <dbl> <fct>
1 essee             0.166 essee
2 muu               0.188 muu
3 essee             0.158 essee
4 muu               0.180 muu
5 muu               0.185 muu
6 muu               0.186 muu
```

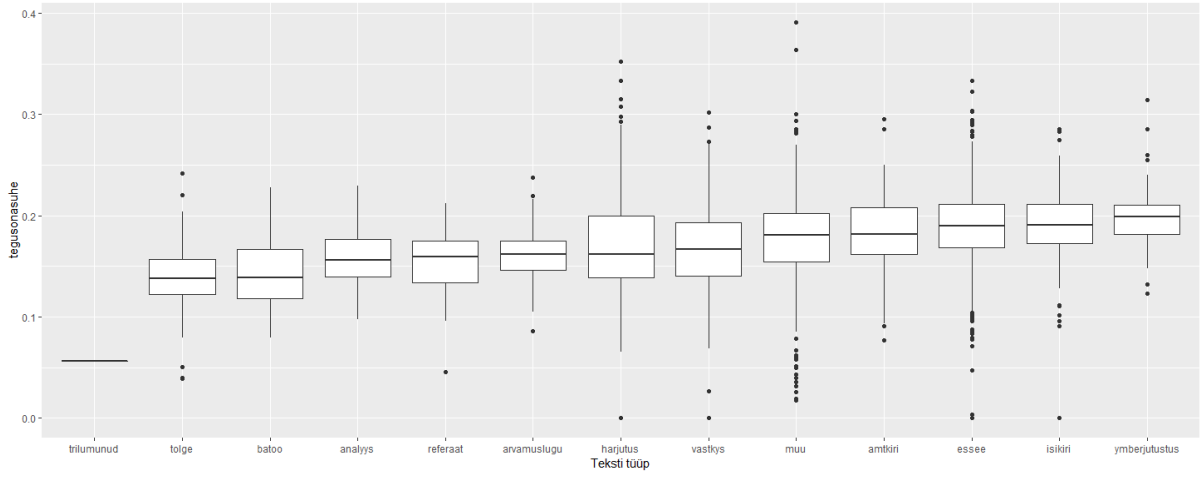
ja nende järgi mediaanosakaalud

```
tyypide_jarjestus=tsuhted %>% group_by(tekstityyp) %>%
   summarise(suhtekeskmine=median(tegusonasuhe)) %>% ungroup() %>% arrange(suhtekeskmine)
```

```
> tyypide_jarjestus
# A tibble: 13 x 2
   tekstityyp    suhtekeskmine
   <chr>                 <dbl>
 1 trilumunud           0.0567
 2 tolge                0.138
 3 batoo                0.139
 4 analyys              0.156
 5 referaat             0.159
 6 arvamuslugu          0.162
 7 harjutus             0.162
 8 vastkys              0.167
 9 muu                  0.181
10 amtkiri              0.182
11 essee                0.190
12 isikiri              0.191
13 ymberjutustus        0.199
```

Kuvamisel tehase tekstitüübist uus faktortüüp, kus tekstitüüpide nimed on eelnevalt arvutatud järjestuses. Nii kuvatakse nad samal moel ka joonisele

```
tsuhted %>% ggplot(aes(factor(tekstityyp, levels=tyypide_jarjestus$tekstityyp),
tegusonasuhe)) + geom_boxplot() + xlab("Teksti tüüp")
```



Eelpool trükitud tabeli järgi paistab, et näiteks isikliku kirja ja harjutuse vahel on erinevuse olulisus

```
11   0.0253  0.0130    0.0375  7.07e-10 isikiri-harjutus
```

Harjutuse ja ametikirja vahel

```
23 -0.0180 -0.0321   -0.00383 2.17e- 3 harjutus-amtkiri
```

Nii on statistilise meetodi abil üldine pilt käes ning edasi saab juba vajadusel sisuliselt lähemalt uurima hakata, et millest erinevused põhjustatud on ning kas ja mida nad näitavad.

Korrelatsioon

Seos andmete vahel. Võtame uurimiseks juba tuttava faili

```
> sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
> head(sonad)
# A tibble: 6 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
4 kungla kuldsel          7             2             2
5 kungla aal              3             2             0
```

Küsime, et kuivõrd on omavahel seotud sõna tähtede arv ning täishäälikute arv sõnas.

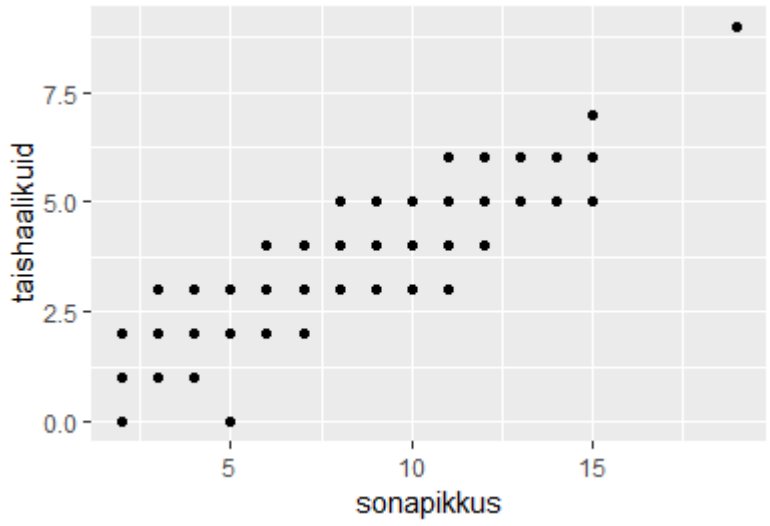
```
> cor(sonad$sonapikkus, sonad$taishaalikuid)
[1] 0.9016922
```

Vastuseks tuleb 0.9 ehk 90% ehk tugevasti.

XY joonisel paistab seos välja nõnda:

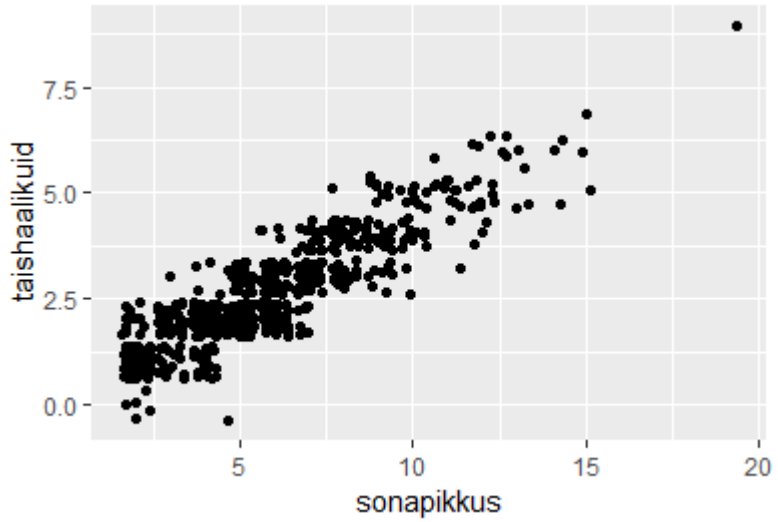
```
> sonad %>% ggplot(aes(sonapikkus, taishaalikuid)) + geom_point()
```

Punktid seisavad üksikutena ridades ja veergudes, kuna tähtede arv sõnas on täisarv



Et aru saada, kui palju kusagil punkte tegelikult on, siis punkte aitab veidi loksutada parameeter `position="jitter"`, nii ei jääda üksteise taha.

```
> sonad %>% ggplot(aes(sonapikkus, taishaalikuid)) + geom_point(position="jitter")
```

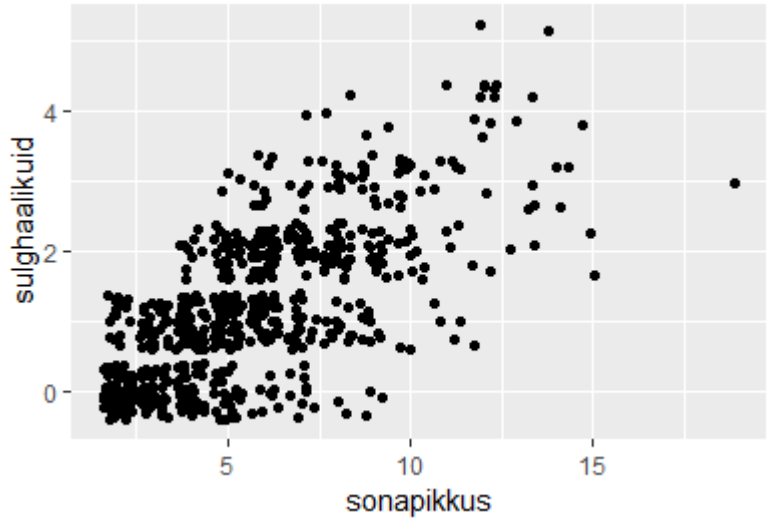


Võrdlusena paistab, et seos sõna pikkuse ning sulghäälikute arvu vahel on märgatavalt nõrgem

```
> cor(sonad$sonapikkus, sonad$sulghaalikuid)
[1] 0.7039988
```

See tuleb välja ka jooniselt:

```
> sonad %>% ggplot(aes(sonapikkus, sulghaalikuid)) + geom_point(position="jitter")
```

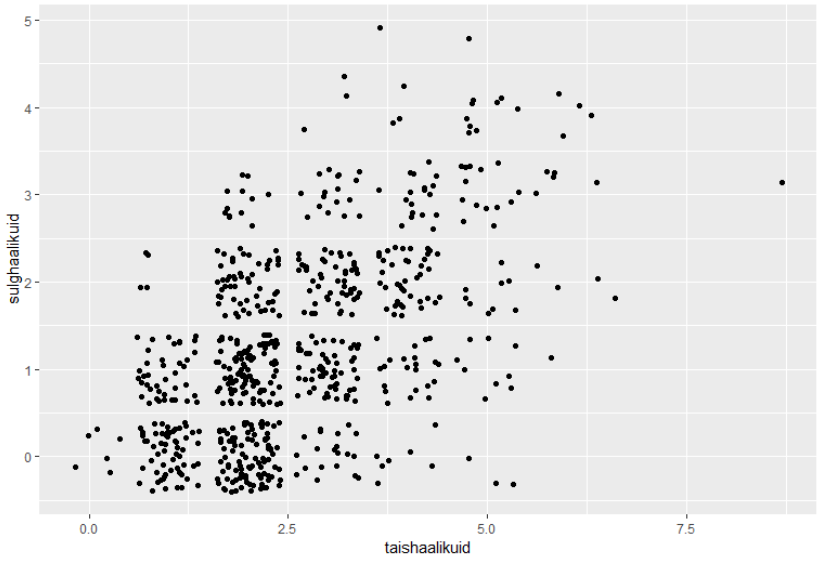


Harjutus

- Tehke näited läbi

- Leidke korrelatsioon täishäälikute ja sulghäälikute arvu vahel
- Koostage illustreeriv joonis
- Leidke korrelatsioon sõnas täishäälikute osakaalu ja sulghäälikute osakaalu vahel, lisage joonis

```
> cor(sonad$taishaalikuid, sonad$sulghaalikuid)
[1] 0.5611331
> sonad %>% ggplot(aes(taishaalikuid, sulghaalikuid)) + geom_point(position="jitter")
```

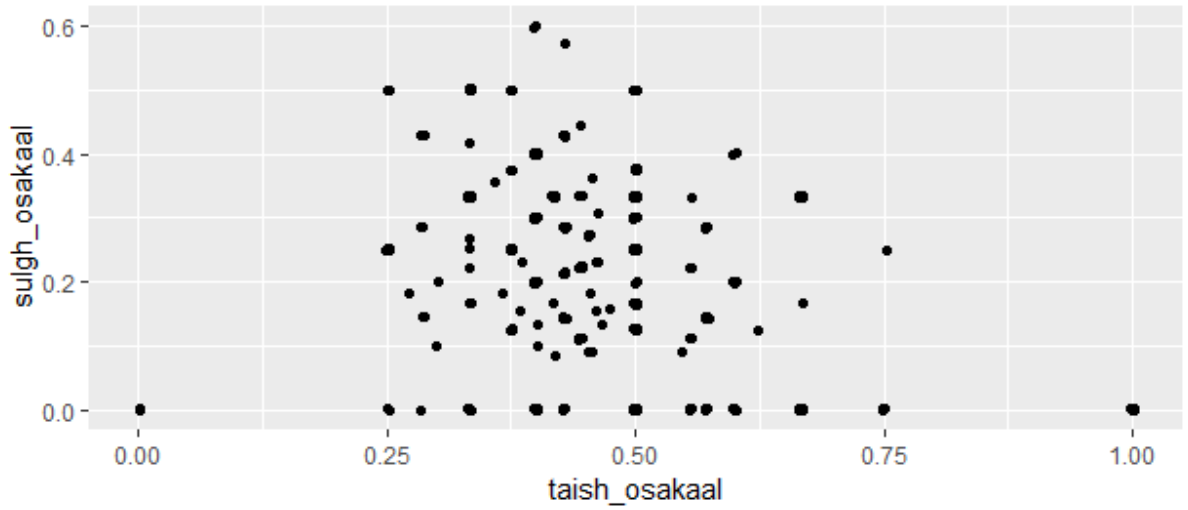


Osakaaludega

```
> osakaaludega=sonad %>% mutate(taish_osakaal=taishaalikuid/sonapikkus,
sulgh_osakaal=sulghaalikuid/sonapikkus)

> head(osakaaludega)
# A tibble: 6 x 7
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid taish_osakaal sulgh_osakaal
  <chr>  <chr>        <int>         <int>         <int>         <dbl>         <dbl>
1 kungla kui              3             2             1         0.667         0.333
2 kungla kungla           6             2             2         0.333         0.333
3 kungla rahvas           6             2             0         0.333         0
4 kungla kuldsel          7             2             2         0.286         0.286
5 kungla aal              3             2             0         0.667         0
6 kungla kord             4             1             2         0.25          0.5
```

Paistab, et täishäälikute ja sulghäälikute osakaalu vahel valitseb nõrk negatiivne seos

```
> cor(osakaaludega$taish_osakaal, osakaaludega$sulgh_osakaal)
[1] -0.2675882
```



Korrelatsioon arvutabelist

Kui tahetakse alles jätta vaid uued arvutatud tulbad, siis võib `mutate` asemel kasutada käsku `transmute` - ehkki `mutate` + `select` annavad vajadusel sama tulemuse.

```
> sonad %>% transmute(taish_osakaal=taishaalikuid/sonapikkus,
sulgh_osakaal=sulghaalikuid/sonapikkus)
# A tibble: 672 x 2
   taish_osakaal sulgh_osakaal
           <dbl>         <dbl>
 1         0.667         0.333
 2         0.333         0.333
 3         0.333         0
 4         0.286         0.286
 5         0.667         0
 6         0.25          0.5
 7         0.4           0.2
 8         0.5           0
 9         0.6           0
10         0.5           0
# ... with 662 more rows
```

Valminud kahetulbalise tabeli saab otse anda `cor`-käsklusele ette

```
> sonad %>% transmute(taish_osakaal=taishaalikuid/sonapikkus,
sulgh_osakaal=sulghaalikuid/sonapikkus) %>% cor()
              taish_osakaal sulgh_osakaal
taish_osakaal     1.0000000    -0.2675882
sulgh_osakaal    -0.2675882     1.0000000
```

Korraga kannatab välja arvutada korrelatsioonid ka rohkemate tulpade vahel - siin kolm tulpa

```
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid)
# A tibble: 672 x 3
```

```
   sonapikkus taishaalikuid sulghaalikuid
        <int>         <int>         <int>
 1          3             2             1
 2          6             2             2
 3          6             2             0
 4          7             2             2
 5          3             2             0
 6          4             1             2
 7          5             2             1
 8          4             2             0
 9          5             3             0
10          4             2             0
# ... with 662 more rows
```

```
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% cor()
              sonapikkus taishaalikuid sulghaalikuid
sonapikkus     1.0000000     0.9016922     0.7039988
taishaalikuid  0.9016922     1.0000000     0.5611331
sulghaalikuid  0.7039988     0.5611331     1.0000000
```

Silma järgi juba võimalik hinnata, et millised seosed on kui tugevad.

Suuremast tabelist arvulised tulbad saab kätte `select_if` käsu abil

```
> sonad %>% select_if(is.numeric)
# A tibble: 672 x 3
   sonapikkus taishaalikuid sulghaalikuid
        <int>         <int>         <int>
 1          3             2             1
 2          6             2             2
 3          6             2             0
 4          7             2             2
 5          3             2             0
 6          4             1             2
 7          5             2             1
 8          4             2             0
 9          5             3             0
10          4             2             0
```

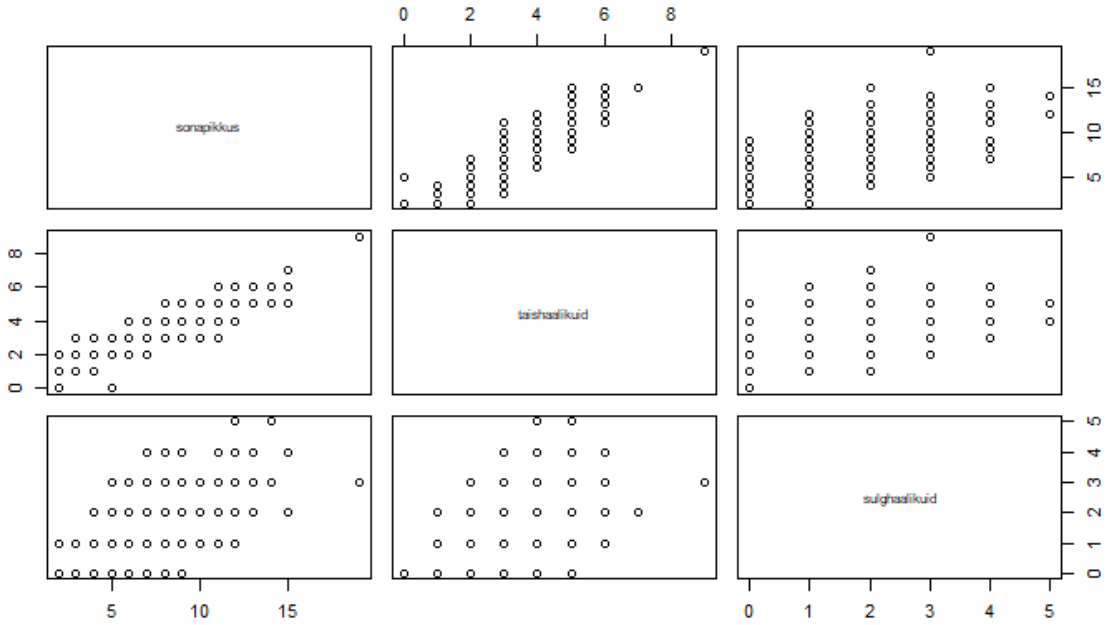
Selle tulemuse saab samuti otse cor-käsklusele saata ja seoseid vaadata.

```
> sonad %>% select_if(is.numeric) %>% cor()
              sonapikkus taishaalikuid sulghaalikuid
sonapikkus     1.0000000     0.9016922     0.7039988
taishaalikuid  0.9016922     1.0000000     0.5611331
sulghaalikuid  0.7039988     0.5611331     1.0000000
```

Joonisena kuvab tulemuse käsklus `pairs`

```
> sonad %>% select_if(is.numeric) %>% pairs()
```

Nii on seosed suures jooniste tabelis näha.



cor.test

Arvutusele lisab kaalu hinnang, et kui tõsiselt arvutust võtta võib. Korrelatsiooni väärtuse usaldusvahemiku annab käsklus cor.test

```
> cor.test(sonad$sonapikkus, sonad$taishaalikuid)

        Pearson's product-moment correlation

data:  sonad$sonapikkus and sonad$taishaalikuid
t = 53.98, df = 670, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8865179 0.9149289
sample estimates:
      cor
0.9016922
```

Käsu väljund lahti seletatult. Sõnapikkuse ja täishäälikute arvu vahel on korrelatsioon 0,9. Praeguse andmestiku põhjal saab seda 95% tõenäosusega üldistada vahemikku 0,87 kuni 0,91. Tõenäosus, et tulpade vahel seos puuduks on 2.2e-16 ehk 0,00000000000000022 ehk liivatera miljoni tonni liiva sees ehk suhteliselt olematu.

Järgmisena vaid Kungla rahva sõnades korrelatsiooni usaldusvahemiku leidmine. Käsklus cor.test on käsuahelas looksulgudesse pandud, kuna me ei soovi, et filtreeritud sõnade tabel läheks tervikuna käsu esimeseks parameetriks, vaid soovime esimeseks parameetriks panna vaid sõnapikkuse tulpa ning teiseks täishäälikute oma.

```
> sonad %>% filter(lugu=="kungla") %>% {cor.test(.$sonapikkus, .$taishaalikuid)}
```

Kuna andmestik on väiksem, siis tuleb 95% usaldusvahemik märgatavalt laiem, ehk siis üldistamisel peaksime arvestama korrelatsiooniga 75% kuni 89%.

```
        Pearson's product-moment correlation

data:  .$sonapikkus and .$taishaalikuid
t = 13.058, df = 73, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7527906 0.8939656
sample estimates:
      cor
0.8367839
```

Sama tulemuse saab kätte ka kahe eraldi käsuna - kui toruahela käsu looksulgudesse panek liiga segane tundub

```
> kunglasonad <- sonad %>% filter(lugu=="kungla")
> cor.test(kunglasonad$sonapikkus, kunglasonad$taishaalikuid)

        Pearson's product-moment correlation

data:  kunglasonad$sonapikkus and kunglasonad$taishaalikuid
t = 13.058, df = 73, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7527906 0.8939656
sample estimates:
      cor
0.8367839
```

Harjutus

- Kuvage korrelatsiooni 95% usaldusvahemik Lambipirni jutu sõnade sõnapikkuse ja täishäälikute arvu vahel
- Leidke tugevamad positiivsed ja negatiivsed seosed sõnaliikide esinemiskordade vahel tekstides. Andmestik http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt , selgitused http://www.tlu.ee/~jaagup/andmed/keel/sonaliikide_lyhendid.txt
  Alusta nimi- ja tegusõnade (S ja V) vahelisest korrelatsioonist.
- Kasuta absoluutarvude asemel osakaale
- Illustreeri tulemusi
- Tuvasta anomaaliaid ning näita võimalusel nende põhjusi. Näita sõnaliikide vahelisi korrelatsiooniseoseid autori keeletaseme ning teksti tüübi kaupa, kasuta ainult eestikeelseid tekste. Illustreeri markantsemaid näiteid. Dokumentide andmed http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt, siduvaks tulbaks "kood"

```
> sonad %>% filter(lugu=="lambipirn") %>% {cor.test(.$sonapikkus, .$taishaalikuid)}
```

```
        Pearson's product-moment correlation

data:  .$sonapikkus and .$taishaalikuid
t = 52.409, df = 595, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8912041 0.9199320
sample estimates:
      cor
0.9066131
```

Korrelatsioon on Lambipirni loos tugevam kui Kungla rahva omas, samas kuna vahemikud kattuvad, siis päris 95% tõenäosusega ei saa veel väita, et seos siinsel juhul tugevam on.

Mugavamaks võrdlemiseks saab usaldusintervalli eraldi välja küsida. Kui usaldusnivoo 90% peale lasta, siis võib juba mõõdetavalt erinevat korrelatsiooni järeldada.

```
> sonad %>% filter(lugu=="lambipirn") %>% {cor.test(.$sonapikkus, .$taishaalikuid,
conf.level=0.9)} %>% .$conf.int
[1] 0.8938339 0.9179207
attr(,"conf.level")
[1] 0.9
> sonad %>% filter(lugu=="kungla") %>% {cor.test(.$sonapikkus, .$taishaalikuid,
conf.level=0.9)} %>% .$conf.int
[1] 0.7684373 0.8862553
attr(,"conf.level")
[1] 0.9
```

Keeleandmed

Tabelis näha tekstide kaupa iga sõnaliigi esinemiskordade arv + lõpus kõikide sõnaüksuste arv

```
> doksonaliigid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt")
```

```
> head(doksonaliigid)
# A tibble: 6 × 18
                 kood     A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z kokku
                <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 doc_100636852915_item    25     0    14     0     3     0    19     5     3    17    54     0    35     0     0    36   211
2 doc_100636852916_item     4     0     5     0     4     0    12     1     3    14    31     0    22     0     0    21   117
3 doc_100636852917_item     9     0     6     0     2     0    13     1     3    17    53     0    25     0     2    27   158
4   doc_1010138197_item    46     7    50     4    20     0    38     3     2    34   183     0   126     0     2   184   699
5   doc_1010138198_item    43     7    49     4    21     0    37     6     2    39   182     0   129     0     2   177   698
6   doc_1010138199_item    45     7    51     4    20     0    38     4     2    37   180     1   132     0     2   185   708
```

Korraga korrelatsiooniseosed sõnaliikide sageduste vahel. Eemaldatud tekstiline kood ning muust eristuv kogusumma. Ümardatud kahe kohani pärast koma, et tulemus korraga ekraanile mahuks.

```
> doksonaliigid %>% select(-kood, -kokku) %>% cor() %>% round(2)
      A     C     D     G     H     I     J     K    N     P     S     U     V     X     Y     Z
A  1.00  0.71  0.89  0.43  0.54 -0.10  0.88  0.83 0.54  0.82  0.69  0.38  0.91  0.30  0.15  0.81
C  0.71  1.00  0.71  0.26  0.35 -0.12  0.72  0.66 0.52  0.65  0.54  0.32  0.72  0.18  0.15  0.62
D  0.89  0.71  1.00  0.42  0.55 -0.05  0.91  0.83 0.56  0.89  0.63  0.39  0.93  0.31  0.16  0.82
G  0.43  0.26  0.42  1.00  0.31 -0.05  0.47  0.30 0.21  0.40  0.39  0.17  0.47  0.17  0.09  0.45
H  0.54  0.35  0.55  0.31  1.00 -0.02  0.56  0.61 0.53  0.47  0.72  0.29  0.51  0.31  0.58  0.76
I -0.10 -0.12 -0.05 -0.05 -0.02  1.00 -0.07 -0.10 0.04 -0.02 -0.16 -0.06 -0.06 -0.03 -0.12 -0.05
J  0.88  0.72  0.91  0.47  0.56 -0.07  1.00  0.82 0.60  0.88  0.67  0.38  0.93  0.29  0.19  0.83
```

```
K  0.83  0.66  0.83  0.30  0.61 -0.10  0.82  1.00 0.59  0.77  0.64  0.39  0.82  0.32  0.20  0.77
N  0.54  0.52  0.56  0.21  0.53  0.04  0.60  0.59 1.00  0.52  0.46  0.19  0.57  0.14  0.29  0.64
P  0.82  0.65  0.89  0.40  0.47 -0.02  0.88  0.77 0.52  1.00  0.57  0.32  0.93  0.27  0.09  0.75
S  0.69  0.54  0.63  0.39  0.72 -0.16  0.67  0.64 0.46  0.57  1.00  0.29  0.66  0.23  0.75  0.86
U  0.38  0.32  0.39  0.17  0.29 -0.06  0.38  0.39 0.19  0.32  0.29  1.00  0.35  0.17  0.08  0.35
V  0.91  0.72  0.93  0.47  0.51 -0.06  0.93  0.82 0.57  0.93  0.66  0.35  1.00  0.29  0.14  0.84
X  0.30  0.18  0.31  0.17  0.31 -0.03  0.29  0.32 0.14  0.27  0.23  0.17  0.29  1.00  0.08  0.29
Y  0.15  0.15  0.16  0.09  0.58 -0.12  0.19  0.20 0.29  0.09  0.75  0.08  0.14  0.08  1.00  0.56
Z  0.81  0.62  0.82  0.45  0.76 -0.05  0.83  0.77 0.64  0.75  0.86  0.35  0.84  0.29  0.56  1.00
```

Sõnaliikide lühendid aadressil

```
http://www.tlu.ee/~jaagup/andmed/keel/sonaliikide_lyhendid.txt
```

```
liigilyhend,liigikirjeldus
A,omadussõna algvõrre
C,omadussõna keskvõrre
D,määrsõna
G,käändumatu omadussõna
H,pärisnimi
I,hüüdsõna
J,sidesõna
K,kaassõna
N,põhiarvsõna
O,järgarvsõna
P,asesõna
S,nimisõna
U,omadussõna ülivõrre
V,tegusõna
X,verbi juurde kuuluv sõna
Y,lühend
Z,lausemärk
```

Ülemises tabelis tulid peaaegu kõik korrelatsioonid positiivsed, sest teksti pikkuse kasvamisega kasvavad ka sõnaliikide esinemise üldarvud. Teksti pikkuse mõju eraldamiseks jagame kõik muud väärtused teksti pikkustega läbi

```
> osakaalud <- doksonaliigid %>% filter(kokku>0) %>% select(-kood) %>% {./.$kokku} %>%
select(-kokku)
> head(round(osakaalud, 3))
      A    C     D     G     H I     J     K     N     P     S     U     V X     Y     Z
1 0.118 0.00 0.066 0.000 0.014 0 0.090 0.024 0.014 0.081 0.256 0.000 0.166 0 0.000 0.171
2 0.034 0.00 0.043 0.000 0.034 0 0.103 0.009 0.026 0.120 0.265 0.000 0.188 0 0.000 0.179
3 0.057 0.00 0.038 0.000 0.013 0 0.082 0.006 0.019 0.108 0.335 0.000 0.158 0 0.013 0.171
4 0.066 0.01 0.072 0.006 0.029 0 0.054 0.004 0.003 0.049 0.262 0.000 0.180 0 0.003 0.263
5 0.062 0.01 0.070 0.006 0.030 0 0.053 0.009 0.003 0.056 0.261 0.000 0.185 0 0.003 0.254
6 0.064 0.01 0.072 0.006 0.028 0 0.054 0.006 0.003 0.052 0.254 0.001 0.186 0 0.003 0.261
```

Leitud korrelatsioonid on nüüd juba nii positiivsed kui negatiivsed

```
> round(cor(osakaalud), 2)
      A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z
A  1.00  0.24  0.21  0.08 -0.30 -0.11  0.39  0.18 -0.20  0.10 -0.29  0.10  0.29  0.02 -0.38 -0.06
C  0.24  1.00  0.06  0.07 -0.24 -0.15  0.31  0.13 -0.21 -0.05 -0.01  0.10  0.07  0.03 -0.05 -0.16
D  0.21  0.06  1.00 -0.08 -0.12  0.08  0.23 -0.11  0.08  0.32 -0.63  0.03  0.49 -0.01 -0.50 -0.05
G  0.08  0.07 -0.08  1.00 -0.01 -0.11  0.15 -0.03 -0.20  0.02 -0.01  0.03  0.03  0.07 -0.05 -0.09
H -0.30 -0.24 -0.12 -0.01  1.00  0.15 -0.46 -0.20  0.20 -0.20  0.02 -0.05 -0.34 -0.01  0.19  0.15
I -0.11 -0.15  0.08 -0.11  0.15  1.00 -0.06 -0.09  0.18  0.12 -0.25 -0.06  0.07 -0.04 -0.14  0.17
J  0.39  0.31  0.23  0.15 -0.46 -0.06  1.00  0.15 -0.22  0.32 -0.38  0.09  0.49  0.06 -0.43 -0.26
K  0.18  0.13 -0.11 -0.03 -0.20 -0.09  0.15  1.00  0.05 -0.01  0.00  0.09  0.05  0.04 -0.20 -0.09
N -0.20 -0.21  0.08 -0.20  0.20  0.18 -0.22  0.05  1.00  0.04 -0.24 -0.13  0.09 -0.04 -0.24  0.12
P  0.10 -0.05  0.32  0.02 -0.20  0.12  0.32 -0.01  0.04  1.00 -0.72  0.01  0.65  0.00 -0.56  0.02
```

```
S -0.29 -0.01 -0.63 -0.01  0.02 -0.25 -0.38  0.00 -0.24 -0.72  1.00 -0.02 -0.79 -0.03  0.67 -0.23
U  0.10  0.10  0.03  0.03 -0.05 -0.06  0.09  0.09 -0.13  0.01 -0.02  1.00  0.03  0.02 -0.02 -0.05
V  0.29  0.07  0.49  0.03 -0.34  0.07  0.49  0.05  0.09  0.65 -0.79  0.03  1.00  0.01 -0.73  0.01
X  0.02  0.03 -0.01  0.07 -0.01 -0.04  0.06  0.04 -0.04  0.00 -0.03  0.02  0.01  1.00 -0.02  0.02
Y -0.38 -0.05 -0.50 -0.05  0.19 -0.14 -0.43 -0.20 -0.24 -0.56  0.67 -0.02 -0.73 -0.02  1.00 -0.17
Z -0.06 -0.16 -0.05 -0.09  0.15  0.17 -0.26 -0.09  0.12  0.02 -0.23 -0.05  0.01  0.02 -0.17  1.00
```

Harjutus

- Leia cor.test- käskluse abil korrelatsioon omadussõnade (A) ja tegusõnade (V) vahel.
- Käivita sama käsklus, kasutades sõnade osakaalu tekstis
- Leia cor.test abil omadussõnade (A) 95% tõenäosusega korrelatsioonivahemikud kõigi teiste sõnaliikidega

```
> cor.test(doksonaliigid$A, doksonaliigid$V)

        Pearson's product-moment correlation

data:  doksonaliigid$A and doksonaliigid$V
t = 251.58, df = 12722, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9095370 0.9153545
sample estimates:
      cor
0.9124919

> cor.test(doksonaliigid$A/doksonaliigid$kokku, doksonaliigid$V/doksonaliigid$kokku)

        Pearson's product-moment correlation

data:  doksonaliigid$A/doksonaliigid$kokku and doksonaliigid$V/doksonaliigid$kokku
t = 33.843, df = 12512, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2734589 0.3055629
sample estimates:
      cor
0.2895923
```

Omadussõna algvõrrete (A) esinemissageduse korrelatsioonid kõigi teiste sõnaliikidega

```
sapply(colnames(doksonaliigid)[3:17], function(sonaliik){
  t=cor.test(doksonaliigid$A, doksonaliigid[[sonaliik]])
  c(t$conf.int[1], t$conf.int[2])
})
```

```
             C         D         G         H           I         J         K         N
[1,] 0.6987612 0.8844591 0.4132350 0.5265258 -0.11642339 0.8809112 0.8259167 0.5244089
[2,] 0.7161180 0.8917978 0.4416358 0.5511846 -0.08201408 0.8884620 0.8366515 0.5491453
             P         S         U         V         X         Y         Z
[1,] 0.8173554 0.6762556 0.3603247 0.9095370 0.2826991 0.1363617 0.8026694
[2,] 0.8285686 0.6946760 0.3901814 0.9153545 0.3143528 0.1702960 0.8146923
```

Peakomponentide analüüs

Kergesti juhtub, et mõõdetavaid tunnuseid tuleb palju, neist arusaadava üldistuse tegemine läheb aga keeruliseks. Tunnuste arvu vähendamiseks levinumad meetodid on peakomponentide analüüs (PCA), faktoranalüüs ning multidimensionaalne skaleerimine (MDS).

Näide kahe tunnusega

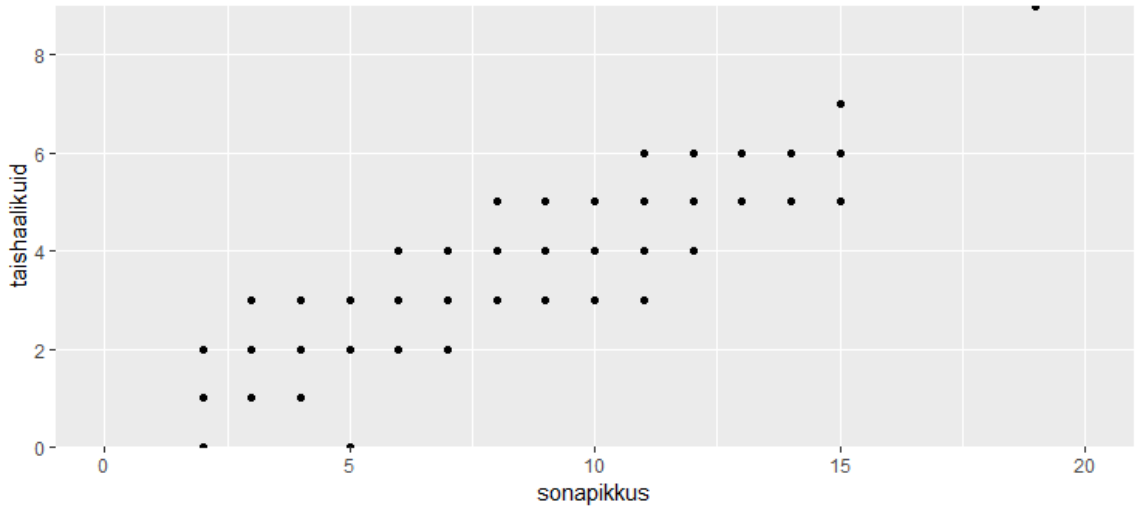
Sisendiks tuttavad sõnad

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Seal sees tulpadena muuhulgas sõnapikkus ja täishäälikute arv

```
> sonad %>% select(sonapikkus, taishaalikuid)
# A tibble: 672 × 2
   sonapikkus taishaalikuid
        <int>         <int>
1           3             2
2           6             2
3           6             2
4           7             2
5           3             2
6           4             1
7           5             2
8           4             2
9           5             3
10          4             2
# ... with 662 more rows
```

```
sonad %>% ggplot(aes(sonapikkus, taishaalikuid)) + geom_point() + coord_equal() + xlim(0,
20) + scale_y_discrete(limits=c(0, 2, 4, 6, 8))
```



Peakomponentide analüüsiks on käsklus `prcomp()`

```
> sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp()
Standard deviations:
[1] 3.0354159 0.5047426

Rotation:
                    PC1        PC2
sonapikkus    0.9221958 -0.3867234
taishaalikuid 0.3867234  0.9221958
```

Vastus lahtiseletatult tähendab, et andmestiku väärtused saab esitada kahe peakomponendi abil. Esimene annab andmete muutusest kätte 3 suhtelist ühikut, teine 0,5.

Lisades käsu `summary()`, arvutatakse tulemused nende standardhälvete ruutude ehk dispersioonide abil - suuremad lähevad suuremaks, väiksemad väiksemaks ning erinevuste ruutude järgi vaadates annab vaid ühe komponendi kasutamine kätte juba 97% kogu andmete täpsusest.

```
> sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% summary()
Importance of components:
                          PC1     PC2
Standard deviation     3.0354 0.50474
Proportion of Variance 0.9731 0.02691
Cumulative Proportion  0.9731 1.00000
```

Vaatame siinse kahetulbalise andmestiku põhjal lähemalt, et milles see komponentideks jagamine siis seisneb

```
> sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation}
                    PC1        PC2
sonapikkus    0.9221958 -0.3867234
taishaalikuid 0.3867234  0.9221958
```

Esimese komponendi väärtus arvestatakse valemi järgi, kus ühe rea (sõna) sõnapikkus korrutatakse 0,92ga ja täishäälikute arv 0,38ga. Et PCA puhul on komponentide suunavektorid risti täisnurga all, siis tulevad teise komponendi puhul samad arvud, ülemisel neist miinusmärk ees.

Graafiliseks kuvamiseks pöörame kõigepealt tabelit `t()` (transpose) käsu abil ning siis muudame ggplotile söödavaks `tibble`-ks

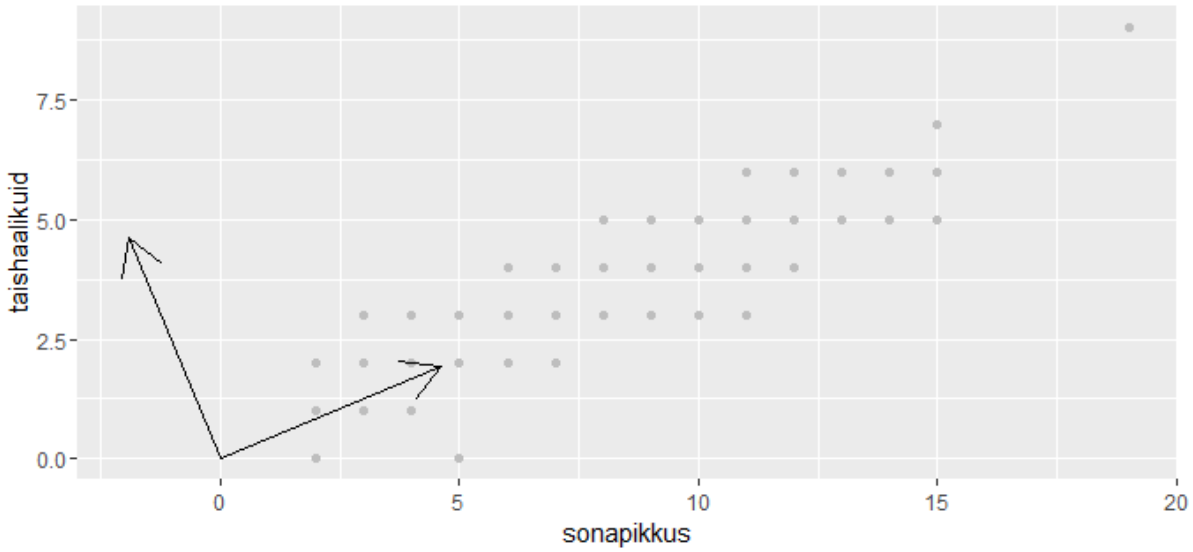
```
> sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation} %>% t()
    sonapikkus taishaalikuid
PC1  0.9221958     0.3867234
PC2 -0.3867234     0.9221958

> sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation} %>% t() %>%
as_tibble()
# A tibble: 2 × 2
  sonapikkus taishaalikuid
       <dbl>         <dbl>
1  0.9221958     0.3867234
2 -0.3867234     0.9221958
```

Salvestame koordinaatide andmed eraldi muutujasse

```
koord <- sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation} %>% t()
%>% as_tibble()
> koord
# A tibble: 2 × 2
  sonapikkus taishaalikuid
       <dbl>         <dbl>
1  0.9221958     0.3867234
2 -0.3867234     0.9221958
```

Kuvame peakomponentide suunad joonisel välja. Paremaks vaatamise proportsiooniks korrutati suundi näitavate noolte pikkused viiega. Esimene peakomponent näitab praegusel juhul teksti pikkust koos keskmise täishäälikusisaldusega, teine täishäälikute sisalduse erinemist keskmisest

```
ggplot() + coord_equal() +
  geom_point(aes(sonapikkus, taishaalikuid), data=sonad, color="gray") +
  geom_segment(data=koord*5, aes(x=0, y=0, xend=sonapikkus, yend=taishaalikuid),
arrow=arrow())
```



Harjutus

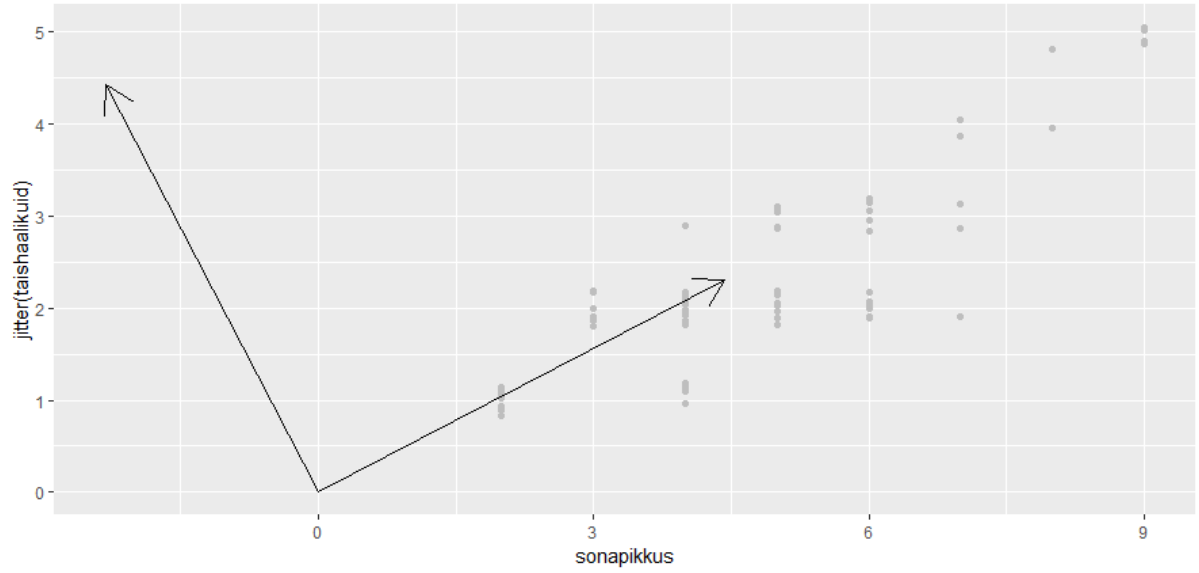
- Tehke näide läbi
- Tehke peakomponentide analüüs täishäälikute ja sõnapikkustega, arvestades ainult Kungla rahva sõnade andmeid. Märgake, millises suunas komponentide koefitsiendid eelmise näitega võrreldes muutusid

- Kuvage uus koordinaatteljestik joonisele.

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
sonad %>% filter(lugu=="kungla") %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% summary()
```

```
Importance of components:
                          PC1    PC2
Standard deviation     2.0368 0.5114
Proportion of Variance 0.9407 0.0593
Cumulative Proportion  0.9407 1.0000
```

```
sonad=sonad %>% filter(lugu=="kungla")
koord <- sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation} %>% t()
%>% as_tibble()
ggplot() + coord_equal() +
  geom_point(aes(sonapikkus, jitter(taishaalikuid)), data=sonad, color="gray") +
  geom_segment(data=koord*5, aes(x=0, y=0, xend=sonapikkus, yend=taishaalikuid),
arrow=arrow())
```



Komponentide väärtuste arvutamine

Nõnda on meil samal joonisel alternatiivne koordinaatteljestik. Arvutame igale sõnale asukoha selles uues koordinaatteljestikus. Avaldis `koord[1, ]` tähendab peakomponentide koordinaatide tabeli esimest rida

```
> koord[1,]
# A tibble: 1 × 2
  sonapikkus taishaalikuid
       <dbl>         <dbl>
1  0.9221958     0.3867234
```

ehk esimese peakomponendi arvestamise koefitsiente

```
> sonad$pc1=sonad$sonapikkus*koord[1, ]$sonapikkus+
    sonad$taishaalikuid*koord[1, ]$taishaalikuid
> head(sonad)
# A tibble: 6 × 6
    lugu    sona sonapikkus taishaalikuid sulghaalikuid      pc1
   <chr>   <chr>      <int>         <int>         <int>    <dbl>
1 kungla     kui          3             2             1 3.540034
2 kungla  kungla          6             2             2 6.306621
3 kungla  rahvas          6             2             0 6.306621
4 kungla kuldsel          7             2             2 7.228817
5 kungla     aal          3             2             0 3.540034
6 kungla    kord          4             1             2 4.075506
```

Sama arvutus ka teise komponendi kohta.

```
> sonad$pc2=sonad$sonapikkus*koord[2, ]$sonapikkus+
   sonad$taishaalikuid*koord[2, ]$taishaalikuid
> head(sonad)
# A tibble: 6 × 7
    lugu    sona sonapikkus taishaalikuid sulghaalikuid      pc1        pc2
   <chr>   <chr>      <int>         <int>         <int>    <dbl>      <dbl>
1 kungla     kui          3             2             1 3.540034  0.6842213
2 kungla  kungla          6             2             2 6.306621 -0.4759489
3 kungla  rahvas          6             2             0 6.306621 -0.4759489
4 kungla kuldsel          7             2             2 7.228817 -0.8626723
5 kungla     aal          3             2             0 3.540034  0.6842213
6 kungla    kord          4             1             2 4.075506 -0.6246979
```

Nagu näha, siis Kungla rahva laulu sõna esimene komponent on suuresti seotud sõna pikkusega, täishäälikute suurem osakaal aga mõnevõrra suurendab selle komponendi arvulist väärtust. Teise komponendi arvulist väärtust aga suurendab täishäälikute osakaal märgatavalt.

```
> sonad %>% filter(sona %in% c("kes", "kui", "uue")) %>% head(3) %>% arrange(pc1)
# A tibble: 3 × 7
       lugu  sona sonapikkus taishaalikuid sulghaalikuid      pc1        pc2
      <chr> <chr>      <int>         <int>         <int>    <dbl>      <dbl>
1 lambipirn   kes          3             1             1 3.153311 -0.2379745
2    kungla   kui          3             2             1 3.540034  0.6842213
3 lambipirn   uue          3             3             0 3.926757  1.6064171
```

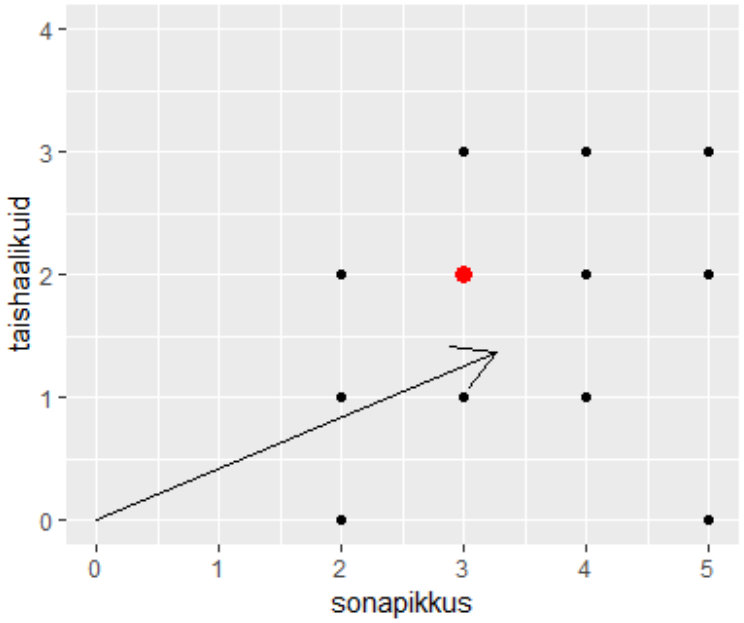
Arvutame esimese sõna esimesele komponendile vastava vektori andmed. Ehk siis algse joonise koordinaadistikku tagasi saamiseks korrutan esimese sõna esimese komponendi väärtuse esimese komponendiga seotud sõnapikkuse koefitsiendiga ning täishäälikute arvu leidmiseks vastava koefitsiendiga.

```
sona=sonad[1, ]
kpc1=koord[1, ]
spc1x<-sona$pc1*kpc1$sonapikkus
spc1y<-sona$pc1*kpc1$taishaalikuid
```

```
> sona
# A tibble: 1 × 7
    lugu  sona sonapikkus taishaalikuid sulghaalikuid      pc1       pc2
   <chr> <chr>      <int>         <int>         <int>    <dbl>     <dbl>
1 kungla   kui          3             2             1 3.540034 0.6842213
> kpc1
# A tibble: 1 × 2
  sonapikkus taishaalikuid
       <dbl>         <dbl>
1  0.9221958     0.3867234
> spc1x
[1] 3.264604
> spc1y
[1] 1.369014
```

Vastavate arvude järgi koostatud joonisel näeb nooleotsa järgi kohta, kuhu paigutataks punkt juhul, kui arvestaksime vaid ühte komponenti

```
sonad %>% ggplot()+coord_equal() +
  geom_segment(x=0, y=0, xend=spc1x, yend=spc1y, arrow=arrow())+
  geom_point(aes(sonapikkus, taishaalikuid)) +
  geom_point(aes(sonapikkus, taishaalikuid), data=sona, color="red", size=3) +
    xlim(0, 5) + ylim(0, 4)
```

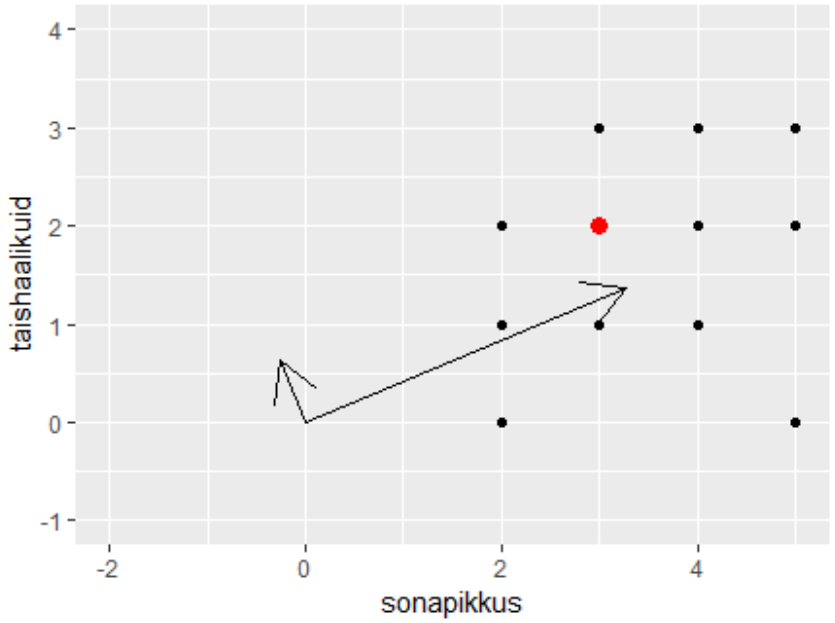


Teise komponendi leidmiseks sarnane tehe

```
kpc2=koord[2, ]
spc2x<-sona$pc2*kpc2$sonapikkus
spc2y<-sona$pc2*kpc2$taishaalikuid

> kpc2
```

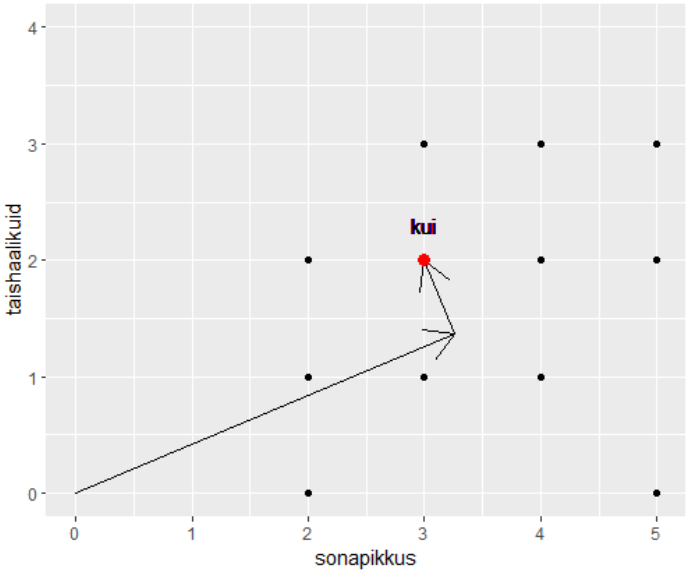
```
# A tibble: 1 × 2
  sonapikkus taishaalikuid
       <dbl>         <dbl>
1 -0.3867234     0.9221958
> spc2x
[1] -0.2646044
> spc2y
[1] 0.630986
```

```
sonad %>% ggplot()+coord_equal() +
  geom_segment(x=0, y=0, xend=spc1x, yend=spc1y, arrow=arrow())+
  geom_segment(x=0, y=0, xend=spc2x, yend=spc2y, arrow=arrow())+
  geom_point(aes(sonapikkus, taishaalikuid)) +
  geom_point(aes(sonapikkus, taishaalikuid), data=sona, color="red", size=3) +
  xlim(-2, 5) + ylim(-1, 4)
```



Kui vektorid üksteise otsa liita, jõutakse ilusasti algsete koordinaatide juurde tagasi

```
sonad %>% ggplot()+coord_equal() +
  geom_segment(x=0, y=0, xend=spc1x, yend=spc1y, arrow=arrow())+
  geom_segment(x=spc1x, y=spc1y, xend=spc1x+spc2x, yend=spc1y+spc2y, arrow=arrow())+
  geom_point(aes(sonapikkus, taishaalikuid)) +
  geom_point(aes(sonapikkus, taishaalikuid), data=sona, color="red", size=3) +
  geom_text(label="kui", x=3, y=2.3)+
  xlim(0, 5) + ylim(0, 4)
```

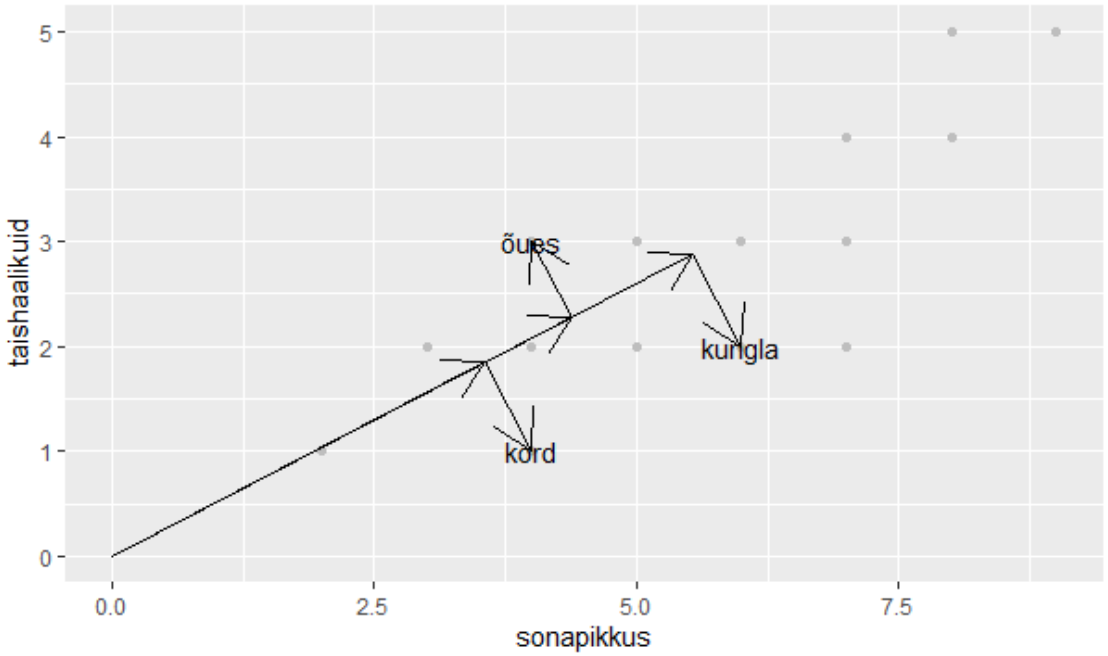


Nii ongi andmeid esitades võimalik valida, et kas meil piisab ühest arvust, et üldist pikkust ja keskmist täishäälikute osakaalu arvestavast väärtusest või on meil konkreetsel juhul ka lisatunnusest kasu.

Harjutus

- Tehke näide läbi
- Lisage uues koordinaatteljestikus nooled ka sõnade "kungla" ning "kord" tarbeks

```
library(tidyverse)
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haali
kud.txt")
sonad=sonad %>% filter(lugu=="kungla")
koord = sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp() %>% {.$rotation} %>% t()
%>% as_tibble()
sonad$pc1=sonad$sonapikkus*koord[1, ]$sonapikkus+
  sonad$taishaalikuid*koord[1, ]$taishaalikuid
sonad$pc2=sonad$sonapikkus*koord[2, ]$sonapikkus+
  sonad$taishaalikuid*koord[2, ]$taishaalikuid
sonad$spc1x=sonad$pc1*koord[1, ]$sonapikkus
sonad$spc1y=sonad$pc1*koord[1, ]$taishaalikuid
head(sonad)
valitud=sonad %>% filter(sona %in% c("õues", "kungla", "kord"))
ggplot() + coord_equal() +
  geom_point(aes(sonapikkus, taishaalikuid), data=sonad, color="gray") +
  geom_segment(aes(x=0, y=0, xend=spc1x, yend=spc1y),
               data=valitud, arrow=arrow()) +
  geom_segment(aes(x=spc1x, y=spc1y, xend=sonapikkus, yend=taishaalikuid),
            data=valitud, arrow=arrow()) +
  geom_text(aes(x=sonapikkus, y=taishaalikuid, label=sona), data=valitud)
```



Peakomponentide tähendus

Teise peakomponendi järgi järjestatuna tulevad ettepoole väiksema täishäälikute osakaaluga sõnad. Esimene neist arv, kus pole ühtegi täishäälikut.

```
> sonad %>% arrange(pc2)
# A tibble: 672 × 7
        lugu            sona sonapikkus taishaalikuid sulghaalikuid       pc1        pc2
       <chr>           <chr>      <int>         <int>         <int>     <dbl>      <dbl>
1  lambipirn           30-40          5             0             0  4.610979 -1.9336170
2  lambipirn     perversselt         11             3             2 11.304324 -1.4873702
3  lambipirn intelligentselt         15             5             4 15.766553 -1.1898723
4  lambipirn      alljärgnev         10             3             1 10.382128 -1.1006468
5  lambipirn      seltskonna         10             3             2 10.382128 -1.1006468
6  lambipirn    tunketaskust         12             4             5 12.613243 -0.9518978
7  lambipirn    kontrollitud         12             4             4 12.613243 -0.9518978
8  lambipirn    mundrikandja         12             4             3 12.613243 -0.9518978
9     kungla         kuldsel          7             2             2  7.228817 -0.8626723
10 lambipirn         kruttis          7             2             3  7.228817 -0.8626723
# ... with 662 more rows
```

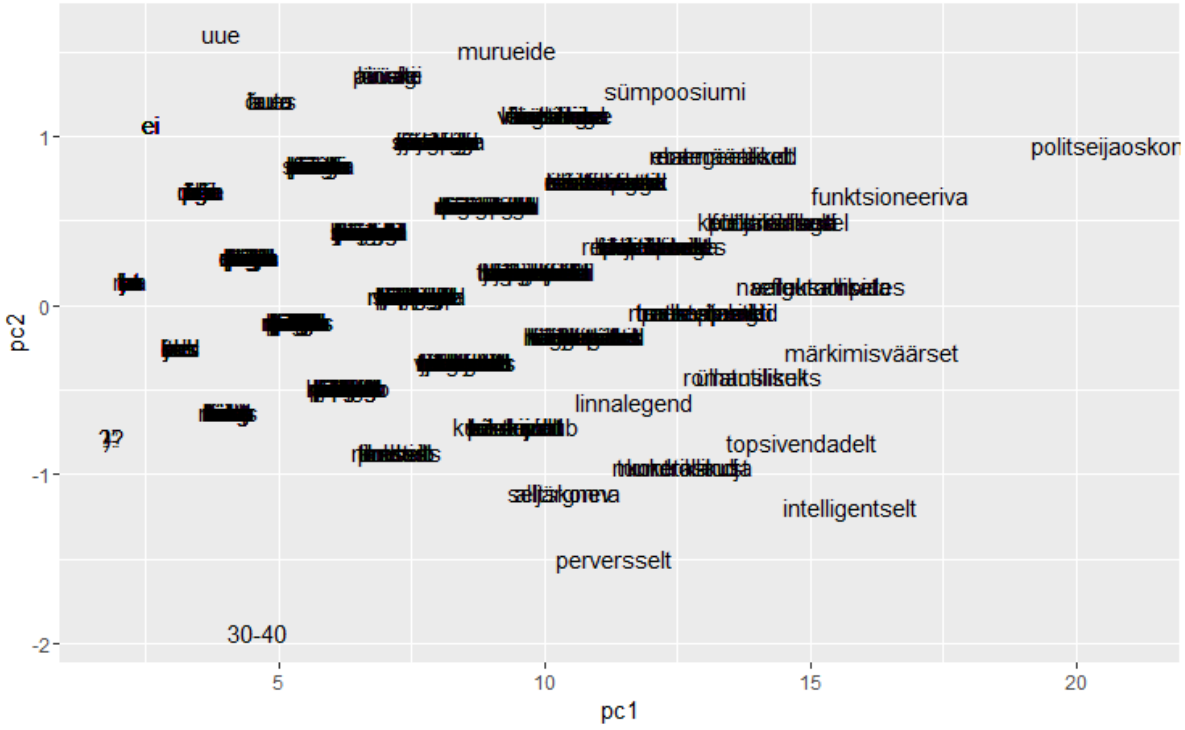
Järjestuse tagaosas jällegi suurema täishäälikute osakaaluga sõnad.

```
> sonad %>% arrange(pc2) %>% tail()
# A tibble: 6 × 7
       lugu     sona sonapikkus taishaalikuid sulghaalikuid      pc1      pc2
      <chr>    <chr>      <int>         <int>         <int>    <dbl>    <dbl>
1 lambipirn   lauale          6             4             0 7.080068 1.368443
2 lambipirn   öösiti          6             4             1 7.080068 1.368443
3 lambipirn   ainuke          6             4             1 7.080068 1.368443
4 lambipirn   püüagi          6             4             2 7.080068 1.368443
```

```
5    kungla murueide          8             5             1 9.311183 1.517192
6 lambipirn      uue          3             3             0 3.926757 1.606417
```

Sama pildi saab joonisel. Vasakult paremale lähevad sõnad pikemaks. Keskel on tavapärase häälikuvahekorraga sõnad, üles ja alla paiknemine sõltub täishäälikute osakaalust

```
sonad %>% ggplot(aes(pc1, pc2, label=sona)) + geom_text()
```



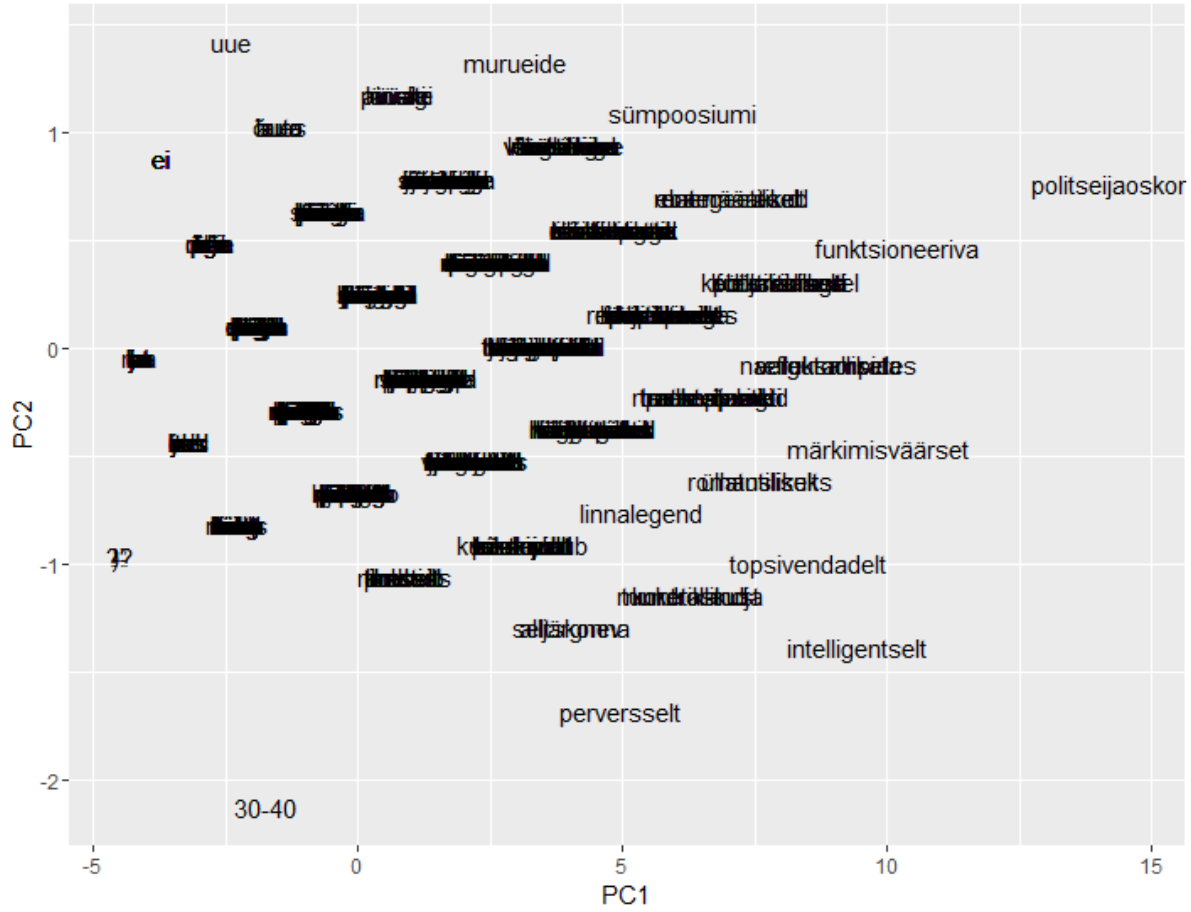
Esimene näide oli pigem meetodi tutvustamiseks - kuigi aitab ka kahe tunnuse puhul nendevahelise seose hästi välja tuua ning näidata, kuhu võiks uue koordinaadistiku teljed paigutada, et ühe arvuga sealset sõna võimalikult hästi iseloomustada saaks.

Sarnased peakomponentide väärtused iga sõna kohta saab ka otse küsida `prcomp`-käskluse `x`-parameetri alt

```
> analyys <- sonad %>% select(sonapikkus, taishaalikuid) %>% prcomp()
> analyys$x %>% head()
             PC1        PC2
[1,] -2.78894032  0.4950410
[2,] -0.02235306 -0.6651292
[3,] -0.02235306 -0.6651292
[4,]  0.89984269 -1.0518526
[5,] -2.78894032  0.4950410
[6,] -2.25346797 -0.8138782
```

Joonis kusjuures tuleb isetehtuga enamjaolt sarnane

```
analyys$x %>% as_tibble() %>% ggplot(aes(PC1, PC2)) + geom_text(label=sonad$sona)
```



X-telje pealt on aga näha, et isetehtud valemi juures hakkas esimese peakomponendi koordinaadi väärtus nullist, siin aga miinus viie juurest. Lähemal uurimisel paistab, et prcomp-käskluse väljundi koordinaadid on nihutatud nõnda, et nende keskmine oleks 0

```
analyys$x[, 1] %>% mean()
[1] 4.448093e-16
```

Sõnade juures arvutatud peakomponentide väärtuste juurest saab käsu väljundi leida andmeid keskmise võrra nihutades

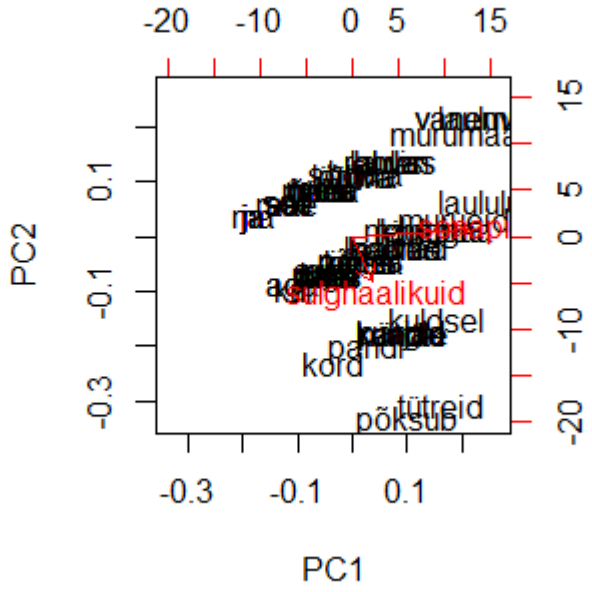
```
> sonad$pc1 %>% mean()
[1] 6.328974
> sonad$pc1 %>% head()
[1] 3.540034 6.306621 6.306621 7.228817 3.540034 4.075506

> sonad$pc1 %>% head() - sonad$pc1 %>% mean()
[1] -2.78894032 -0.02235306 -0.02235306  0.89984269 -2.78894032 -2.25346797
> analyys$x[, 1] %>% head()
[1] -2.78894032 -0.02235306 -0.02235306  0.89984269 -2.78894032 -2.25346797
```

Harjutus

Andmestikuks võtke ainult Kungla rahva sõnad

- Koostage XY joonis sõnapikkuste ja **sulg**häälikutega - kord punktidena, kord sõnadega (geom_text())
- Leidke analüüsi abil peakomponendid, vaadake kuivõrd kumbki sõna üldist asukohta teljestikul määrab
- Arvutage sõnadele koordinaadid uues, kahe peakomponendi teljestikus
- Kuvage esimene peakomponent noolena teljestikule
- Vaadake laulu paari sõna järgi, kuidas vastava sõnani jõutakse uute peakomponentide kaudu
- Kuva laulu esimese sõnani jõudmine kahe peakomponendi noolte abil
- Joonista biplot näitamaks sõnade ja komponentide paigutust

```
 koord = sonad %>% select(sonapikkus, sulghaalikuid) %>% prcomp() %>%
biplot(xlabs=sonad$sona)
```



Kolm tunnust

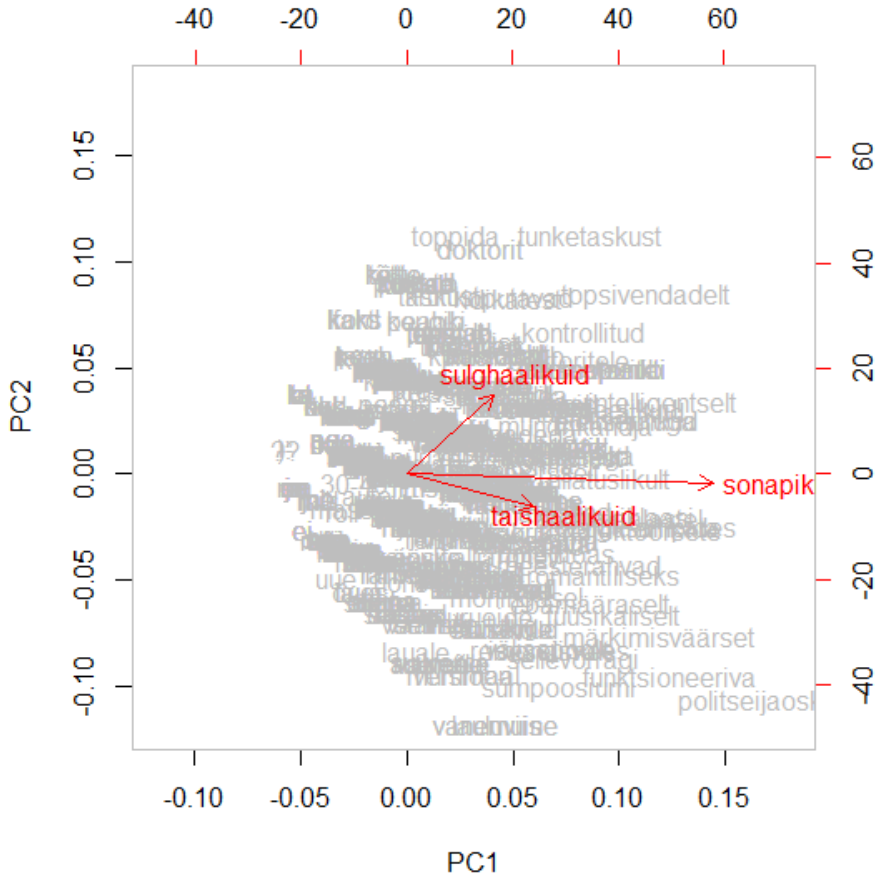
Sulghäälikuid lisades saab PCA ka kolm komponenti. Rotatsioone ehk laadumisi vaadates paistab, et esimene neist on enim seotud pikkusega, teine sulghäälikutega ja kolmas täishäälikutega, ent iga komponent on siiski kõigist mõjutatud. Ning sõna paiknemise koha pealt kolme mõõtmega ruumis annab esimene neist esimene 92%, teine 6% ja kolmas 2%.

```
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% prcomp()
Standard deviations:
[1] 3.1314304 0.7810933 0.4660785

Rotation:
```

```
                    PC1         PC2         PC3
sonapikkus    0.8933502 -0.1036315 -0.4372480
taishaalikuid 0.3711984 -0.3782049  0.8480406
sulghaalikuid 0.2532530  0.9199030  0.2994016
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% prcomp() %>% summary()
Importance of components:
                          PC1     PC2     PC3
Standard deviation     3.1314 0.78109 0.46608
Proportion of Variance 0.9222 0.05738 0.02043
Cumulative Proportion  0.9222 0.97957 1.00000
```

Sõnade seotuse peakomponentidega + üksikute tunnuste veetud suunad joonisel saab välja joonistada käsuga biplot()

```
sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% prcomp() %>%
 biplot(col=c("gray", "red"), xlabs=sonad$sona)
```

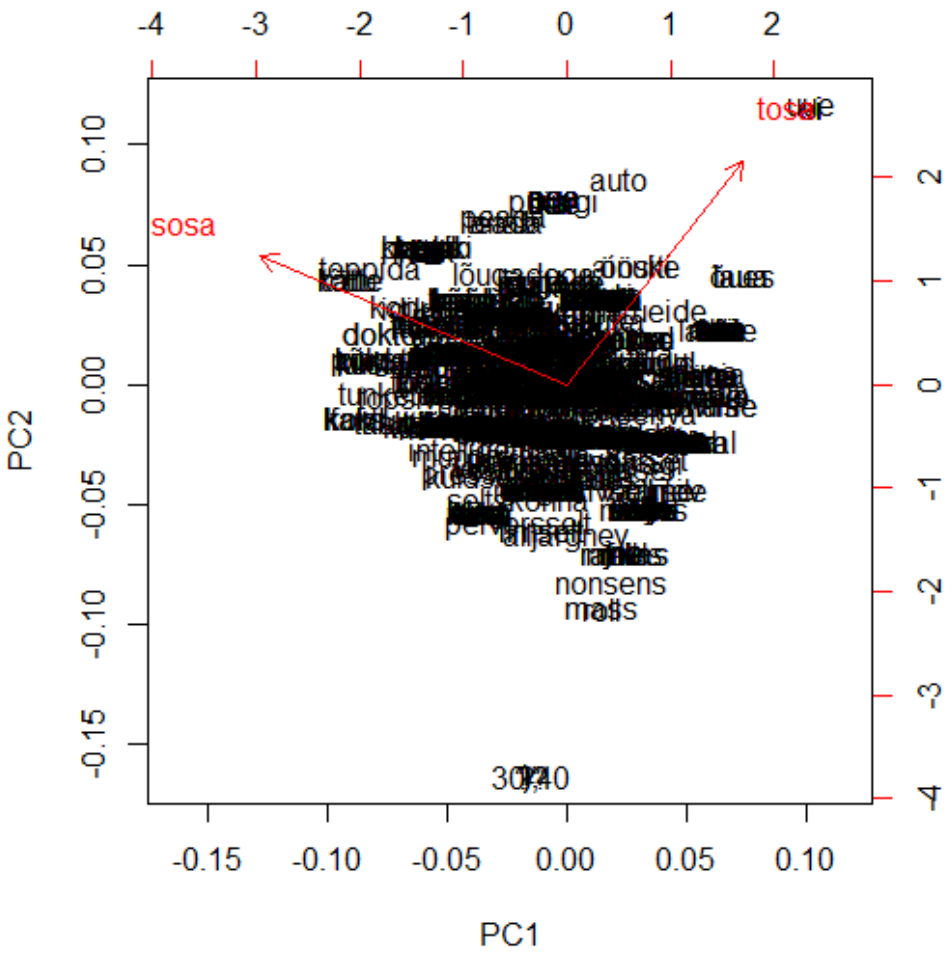


Harjutus

- Leidke täishäälikute ja sulghäälikute osakaal sõnas. Tehke nende osakaalude põhjal peakomponentide analüüs ning joonistage biplot()

```
> sonad %>% transmute(tosa=taishaalikuid/sonapikkus, sosa=sulghaalikuid/sonapikkus)
# A tibble: 672 x 2
    tosa  sosa
   <dbl> <dbl>
 1 0.667 0.333
 2 0.333 0.333
 3 0.333 0
 4 0.286 0.286
 5 0.667 0
 6 0.25  0.5
 7 0.4   0.2
 8 0.5   0
 9 0.6   0
10 0.5   0

> sonad %>% transmute(tosa=taishaalikuid/sonapikkus, sosa=sulghaalikuid/sonapikkus) %>%
prcomp()
Standard deviations (1, .., p=2):
[1] 0.1640490 0.1195324

Rotation (n x k) = (2 x 2):
            PC1       PC2
tosa  0.4971834 0.8676455
sosa -0.8676455 0.4971834
> sonad %>% transmute(tosa=taishaalikuid/sonapikkus, sosa=sulghaalikuid/sonapikkus) %>%
prcomp() %>% summary()
Importance of components:
                          PC1    PC2
Standard deviation     0.1640 0.1195
Proportion of Variance 0.6532 0.3468
Cumulative Proportion  0.6532 1.0000
```

```
sonad %>% transmute(tosa=taishaalikuid/sonapikkus, sosa=sulghaalikuid/sonapikkus) %>%
prcomp() %>% biplot(xlabs=sonad$sona)
```



Hulk sõnaliike

Enamasti läheb peakomponentanalüüsi vaja oludes, kus tunnuseid on palju rohkem - nagu näites, kus loendatakse iga teksti kohta mitukümmend sõnaliiki

```
> doksonaliigid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt")
```

```
> head(doksonaliigid)
# A tibble: 6 x 18
  kood                    A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z kokku
  <chr>               <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 doc_100636852915_item    25     0    14     0     3     0    19     5     3    17    54     0    35     0     0    36   211
2 doc_100636852916_item     4     0     5     0     4     0    12     1     3    14    31     0    22     0     0    21   117
3 doc_100636852917_item     9     0     6     0     2     0    13     1     3    17    53     0    25     0     2    27   158
4 doc_1010138197_item      46     7    50     4    20     0    38     3     2    34   183     0   126     0     2   184   699
5 doc_1010138198_item      43     7    49     4    21     0    37     6     2    39   182     0   129     0     2   177   698
6 doc_1010138199_item      45     7    51     4    20     0    38     4     2    37   180     1   132     0     2   185   708
```

Jätame alles vaid iga sõnaliigi sagedused ning vaatame, mida analüüs näitab

Nagu näha, siis esimene komponent annab juba üle 80% tekstiga seotud üldandmetest - kahtlustada küll võib suurt seost teksti pikkusega.

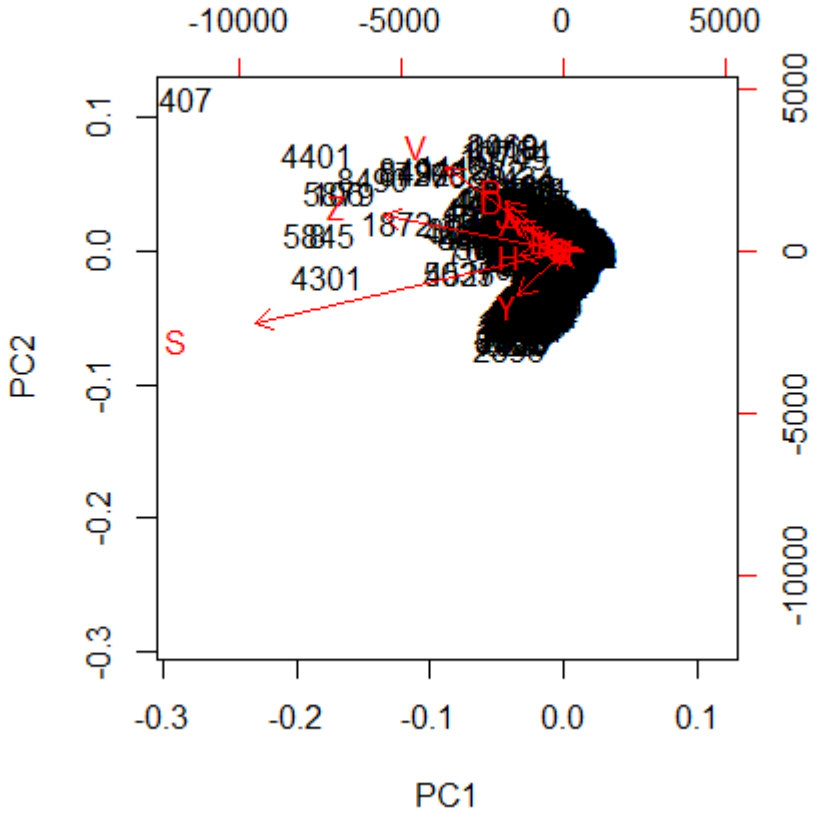
```
> doksonaliigid %>% select(-kood, -kokku) %>% prcomp() %>% summary()
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13    PC14    PC15    PC16
Standard deviation     135.0486 49.7028 22.18392 11.80881 10.80418 8.58053 8.3447 6.93826 6.19669 5.40773 3.16148 2.44456 1.4469 0.82345 0.38386 0.1987
Proportion of Variance   0.8391  0.1137  0.02264  0.00642  0.00537 0.00339 0.0032 0.00221 0.00177 0.00135 0.00046 0.00027 0.0001 0.00003 0.00001 0.0000
Cumulative Proportion    0.8391  0.9528  0.97542  0.98184  0.98721 0.99060 0.9938 0.99602 0.99778 0.99913 0.99959 0.99986 1.0000 0.99999 1.00000 1.0000
```

Lähemalt laadumisi uurides näeb, et esimene komponent on suurelt jaolt seotud nimi- ja tegusõnade arvu ning kirjavahemärkidega (S, V, Z). Teises komponendis tulevad lisaks esile lühendid (Y) ning kolmandas taas kirjavahemärgid.

```
> doksonaliigid %>% select(-kood, -kokku) %>% prcomp()
Standard deviations (1, .., p=16):
 [1] 135.0485909  49.7028457  22.1839163  11.8088128  10.8041828   8.5805270   8.3447212
6.9382582   6.1966915   5.4077308   3.1614801   2.4445601
[13]   1.4468744   0.8234489   0.3838606   0.1987394

Rotation (n x k) = (16 x 16):
           PC1           PC2           PC3           PC4           PC5           PC6           PC7           PC8           PC9          PC10
A -0.1042358153  0.1617596550 -1.152626e-01  0.131677465  0.3251000529 -0.1605951342 -0.0424483572 -0.0091042416  0.2061748730  0.8573340644
C -0.0106128346  0.0165646065 -1.438056e-02 -0.005964216  0.0121973276 -0.0524811144  0.0310814361 -0.0033224839 -0.0026990550  0.0075506209
D -0.1453630253  0.2822204931 -5.165072e-02  0.197117040 -0.0703188433 -0.6841956366 -0.4865278076  0.0922600139  0.2143169688 -0.2957568769
G -0.0113402575  0.0133376520 -6.835461e-03  0.001250505  0.0575819544  0.0648181568 -0.0137814623 -0.0012332022 -0.0962049570 -0.1136374071
H -0.1091113742 -0.0279843331  3.325376e-01  0.869846901 -0.0902395853  0.2321690624  0.0395323966  0.2183051159 -0.0683663934 -0.0025755172
I  0.0008175751  0.0020816469  7.401733e-03  0.001409184 -0.0020133468  0.0061939057  0.0002635283 -0.0091207316  0.0081267936 -0.0139860653
J -0.1219314326  0.2071374939 -6.556711e-02  0.114063130  0.0243003565 -0.2891331630  0.1431064657 -0.3604282496 -0.8257134845  0.0630058096
K -0.0332003099  0.0477377908 -9.091934e-03  0.102288946  0.0345676931 -0.0710656789  0.0303853961 -0.0349609414  0.0634030907  0.0406157687
N -0.0456856564  0.0609975513  1.569059e-01  0.067899843 -0.0336733331 -0.3661027089  0.7957477412 -0.1968823297  0.3589879587 -0.1289926070
P -0.1461539910  0.3392883483 -1.566023e-01  0.042936079 -0.6895731965  0.2769212051 -0.0954464222 -0.4623205449  0.1999231600  0.1389322195
S -0.7795029004 -0.4913478507 -3.532117e-01  0.044219057  0.0689980077  0.0289439217  0.0153960594 -0.0848187108  0.0522915051 -0.0895347627
U -0.0010627337  0.0014495124  9.489718e-05  0.004438942  0.0029629965 -0.0038315060 -0.0055431474 -0.0017068207 -0.0024859023  0.0003815372
V -0.2968219885  0.5800620142 -2.787985e-01 -0.131093239 -0.0076923394  0.1821398640  0.2669295142  0.5999920911 -0.0892698499 -0.0795182216
X -0.0004346759  0.0006144668  5.176807e-04  0.003253975  0.0005374024  0.0008490438 -0.0023727824  0.0006095124 -0.0009948014  0.0008778177
Y -0.1150715906 -0.3022923448  2.951310e-01 -0.229523800 -0.5886021153 -0.3044412064  0.0456724791  0.4083735715 -0.1763742997  0.3281081600
Z -0.4554873595  0.2442248110  7.280399e-01 -0.292153407  0.2191839147  0.1382448582 -0.1463967528 -0.1614361217  0.0327805751 -0.0278805882
           PC11          PC12          PC13          PC14          PC15          PC16
A  0.145627252 -0.001282697 -0.012125236 -0.0095748592 -7.093308e-04 -0.0008811507
C -0.071320298 -0.050969948  0.991317574 -0.0678150739 -2.060099e-02  0.0027996669
D  0.065855457 -0.035223305 -0.019633152  0.0046960001 -4.379933e-03 -0.0006486105
G  0.726290761  0.658841991  0.087983015 -0.0211708006 -1.095227e-03 -0.0025274388
H  0.025775764 -0.054819593  0.021332368  0.0036334952 -2.122740e-03 -0.0028591099
I  0.013075841 -0.038879212 -0.068391440 -0.9964582551  1.523742e-02  0.0054501324
J -0.031654519 -0.048889025 -0.034074590 -0.0023395224 -3.087405e-03 -0.0003800828
K -0.651693070  0.738540581 -0.017742545 -0.0364781621 -1.377320e-02 -0.0056178356
N  0.102246923 -0.022870440 -0.034436729  0.0105851632  5.144710e-03  0.0025959831
P  0.044977465  0.020207233  0.019466410  0.0049709945  1.176329e-03 -0.0001066909
S -0.002242913 -0.020549669 -0.005508941 -0.0005704547  4.204416e-05  0.0002731113
U -0.010060018  0.009921296  0.021403130  0.0132024749  9.994787e-01 -0.0111319587
V -0.032875273 -0.018325681 -0.017462054 -0.0036481545  2.761483e-03  0.0001914170
X -0.001843701  0.006125787 -0.001919750  0.0054925498  1.099663e-02  0.9998920450
Y  0.029761624  0.088296124  0.000672261 -0.0124018944  1.847471e-03  0.0002637059
Z -0.027858514 -0.010779120  0.006731119  0.0072197694 -1.140506e-03 -0.0002008867
```

Joonise abil paistab, et kuhu poole üksikud tunnusel tekste suunavad

```
doksonaliigid %>% select(-kood, -kokku) %>% prcomp() %>% biplot()
```



Küllalt palju aga sõltub siin arvatavasti teksti pikkusest. Pikkuse mõju eemaldamiseks jagame kõik tulbad läbi tulbaga `kokku`.

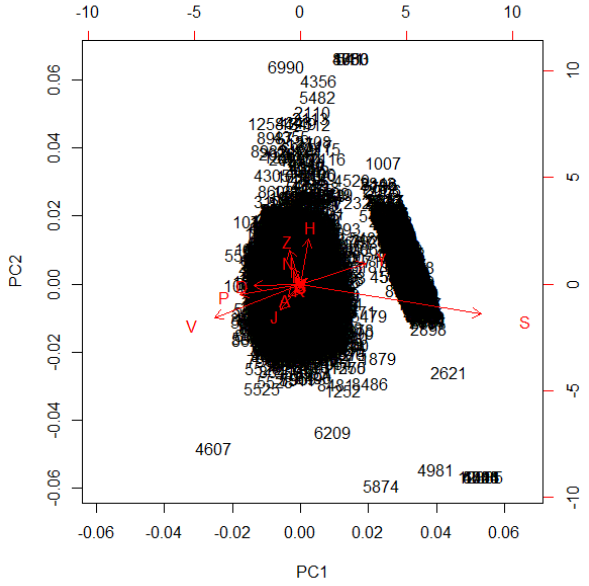
```
> doksonaliigid %>% select(-kood) %>% {./.$kokku} %>% round(3) %>% head()
      A    C     D     G     H I     J     K     N     P     S     U     V X     Y     Z
kokku
1 0.118 0.00 0.066 0.000 0.014 0 0.090 0.024 0.014 0.081 0.256 0.000 0.166 0 0.000 0.171
1
2 0.034 0.00 0.043 0.000 0.034 0 0.103 0.009 0.026 0.120 0.265 0.000 0.188 0 0.000 0.179
1
3 0.057 0.00 0.038 0.000 0.013 0 0.082 0.006 0.019 0.108 0.335 0.000 0.158 0 0.013 0.171
1
4 0.066 0.01 0.072 0.006 0.029 0 0.054 0.004 0.003 0.049 0.262 0.000 0.180 0 0.003 0.263
1
5 0.062 0.01 0.070 0.006 0.030 0 0.053 0.009 0.003 0.056 0.261 0.000 0.185 0 0.003 0.254
1
6 0.064 0.01 0.072 0.006 0.028 0 0.054 0.006 0.003 0.052 0.254 0.001 0.186 0 0.003 0.261
1
```

Peakomponentide analüüsi järgi paistab nüüd, et esimene komponent annab vaid ⅔ teksti üldandmetest ning 90%ni jõudmiseks läheb vaja juba viite komponenti

```
> doksonaliigid %>% select(-kood) %>% {./.$kokku} %>% na.omit() %>% select(-kokku) %>%
prcomp() %>% summary()
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13    PC14    PC15     PC16
Standard deviation     0.1189 0.04394 0.03353 0.03272 0.02756 0.02497 0.02331 0.02074 0.01654 0.01130 0.01071 0.007287 0.00429 0.001247 0.0008905 1.898e-16
Proportion of Variance 0.6667 0.09100 0.05299 0.05044 0.03580 0.02937 0.02561 0.02027 0.01290 0.00602 0.00540 0.002500 0.00087 0.000070 0.0000400 0.000e+00
Cumulative Proportion  0.6667 0.75771 0.81070 0.86114 0.89694 0.92631 0.95193 0.97220 0.98509 0.99112 0.99652 0.999020 0.99989 0.999960 1.0000000 1.000e+00
```

Samuti on joonis märgatavalt sümmeetrilisem

```
doksonaliigid %>% select(-kood) %>% {./.$kokku} %>% na.omit() %>% select(-kokku) %>%
prcomp() %>% biplot()
```



Paistab, et hulk tekste koondub paremas servas ühte, alumise järjekorranumber neist 2898. Uurime tabelist, et mis on selle teksti kood

```
> doksonaliigid[2898, "kood"]
# A tibble: 1 x 1
  kood
  <chr>
1 doc_438618349703_item
```

Ja vaatame, milline on vastava koodiga tekst ise

```
http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_438618349703_item.txt
```

Selgub, et venekeelne

```
СМИ расшифровывается как средство массовой информации.
СМИ бывает разным, это в основном интернет, газеты, телевидение, журналы.
Во всех этих источников информации могут говорить правду или лгать.
```

Metaandmed joonisel

Niisama peakomponentide järgi tabeli ridade numbreid ekraanile joonistades saab heal juhul rühmi leida. Sisulisemate seoste tarbeks tasub vaadata metaandmeid ehk tabelirea muid tulpasid. Keeleandmete puhul leiab näiteks eraldi failist andmed teksti autorite kodukeele, emakeele, keeletaseme, elukoha, vanuse ja muu kohta.

```
dokmeta=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt")
```

Ühine tekstikoodi näitav tulp võimaldab read kõrvuti panna

```
> koos=dokmeta %>% inner_join(doksonaliigid)
Joining, by = "kood"
```

Kümme juhuslikku rida tabelist

```
> koos %>% sample_n(10) %>% select(emakeel, sugu, S, V)
# A tibble: 10 x 4
   emakeel sugu      S     V
   <chr>   <chr> <int> <int>
 1 NA      naine    30    15
 2 NA      mees     17    14
 3 vene    naine    39    40
 4 vene    naine    21    20
 5 vene    mees     42    36
 6 NA      naine    19    21
 7 soome   naine    50    37
 8 NA      NA      278   138
 9 NA      NA       58    33
10 NA      mees     34    15
```

Edasiseks arvutuseks eemaldame tühjade väärtustega read, jätame alles vaid eestikeelsed tekstid ning eemaldame tühjad tekstid

```
> koos <- koos %>% na.omit() %>% filter(tekstikeel=="eesti") %>% filter(kokku>0)
> koos %>% sample_n(10) %>% select(emakeel, sugu, S, V)
# A tibble: 10 x 4
   emakeel sugu      S     V
   <chr>   <chr> <int> <int>
 1 vene    mees     39    39
 2 vene    mees     42    26
 3 vene    mees     38    36
 4 vene    naine    46    51
 5 vene    mees     37    44
 6 saksa   naine    52    43
 7 vene    naine     2     4
 8 vene    naine    37    39
 9 vene    naine    26    29
10 vene    naine    14    13
```

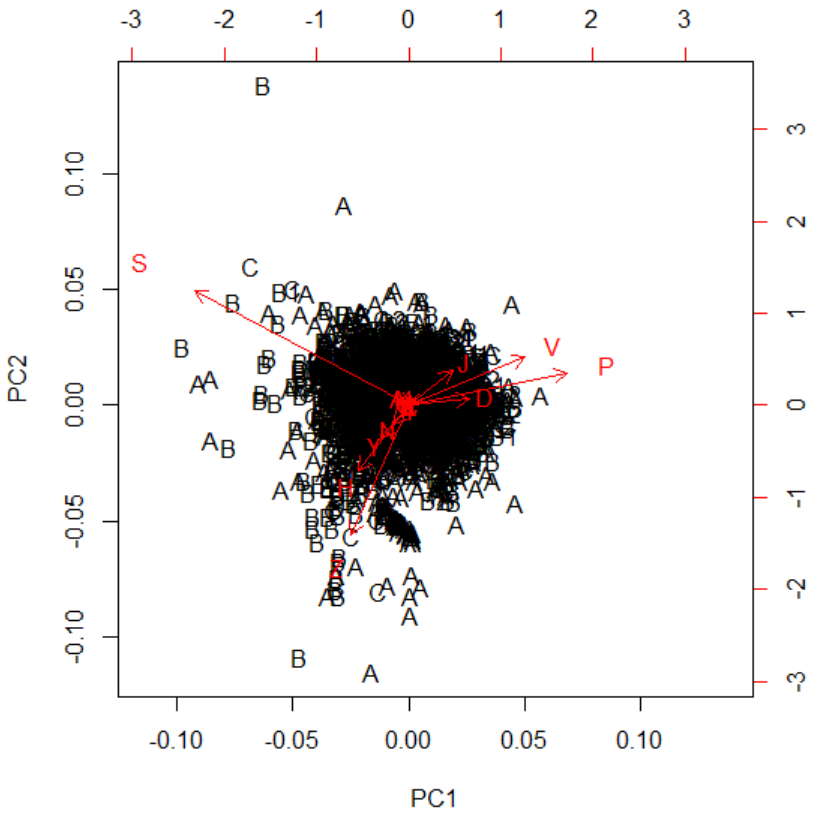
Teksti pikkuse mõju eemaldame jagades kõik arvulised tulbad läbi tulbaga `kokku`

```
koos <- koos %>% mutate_if(is.numeric, funs(./koos$kokku))
```

```
> koos %>% sample_n(10) %>% select(emakeel, sugu, S, V)
# A tibble: 10 x 4
   emakeel sugu      S     V
   <chr>   <chr> <dbl> <dbl>
 1 vene    naine 0.144 0.239
 2 soome   naine 0.190 0.190
 3 soome   naine 0.285 0.162
 4 vene    naine 0.149 0.189
 5 vene    naine 0.254 0.184
 6 vene    naine 0.343 0.139
 7 vene    naine 0.168 0.218
 8 vene    naine 0.214 0.183
 9 vene    naine 0.255 0.182
10 vene    mees  0.215 0.179
```

Edasi tuleb arvuliste tulpade põhjal peakomponentide analüüs ning selle tulemuste esitlemine biploti abil. Nähtavaks kihiks jäetakse iga teksti keeletase.

```
koos  %>% select_if(is_numeric) %>% select(-kokku) %>% prcomp() %>%
biplot(xlabs=koos$keeletase)
```

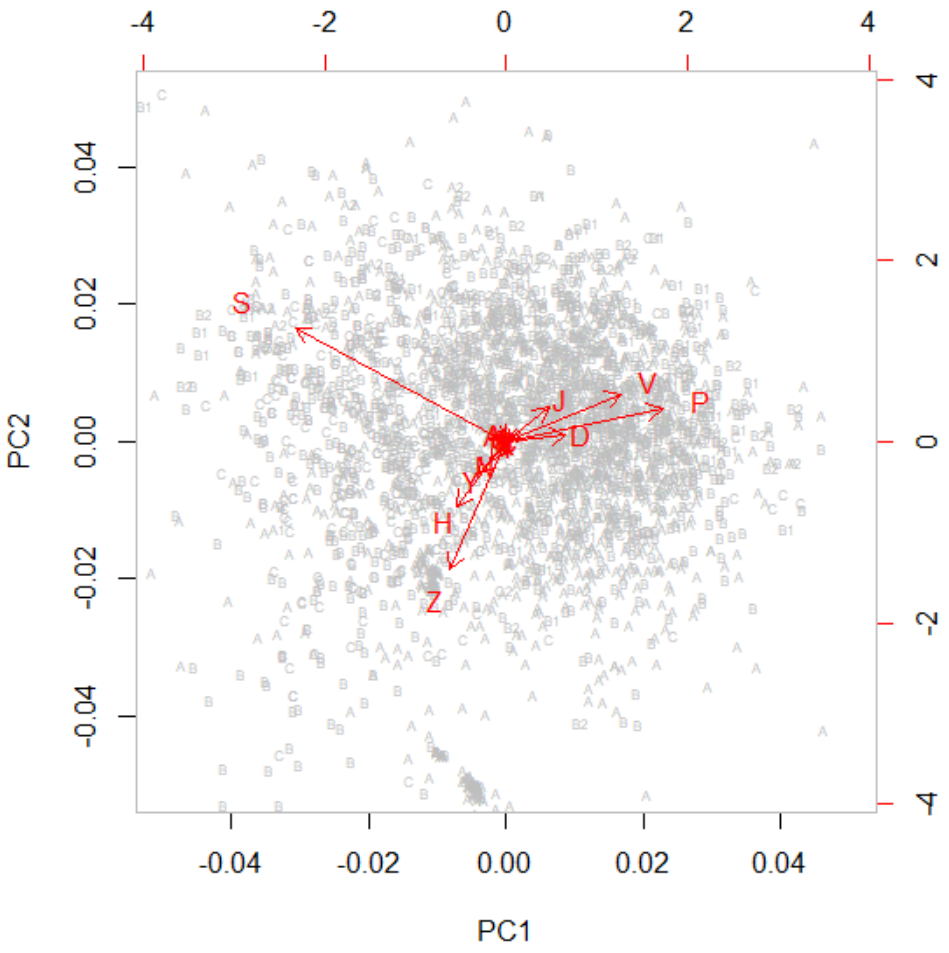


Selget rühmitumist ei paista, küll aga on A-taseme ehk algajate tekste igal pool äärmuste juures - see annab vihje, et sealsed andmed võivad rohkem kõikuda.

Joonise mugavamaks lugemiseks mõned käsklused juurde - cex teksti suuruste jaoks: 0.5 ühikut keeletasemete kohta ning 1 ühik ehk muutmata kuju PCA moodustanud sõnaliikide kuvamiseks. Parameetrite xlim ning ylim abil saab suurendada välja joonise keskosa, värvid ka vastavalt kummagi andmestiku jaoks

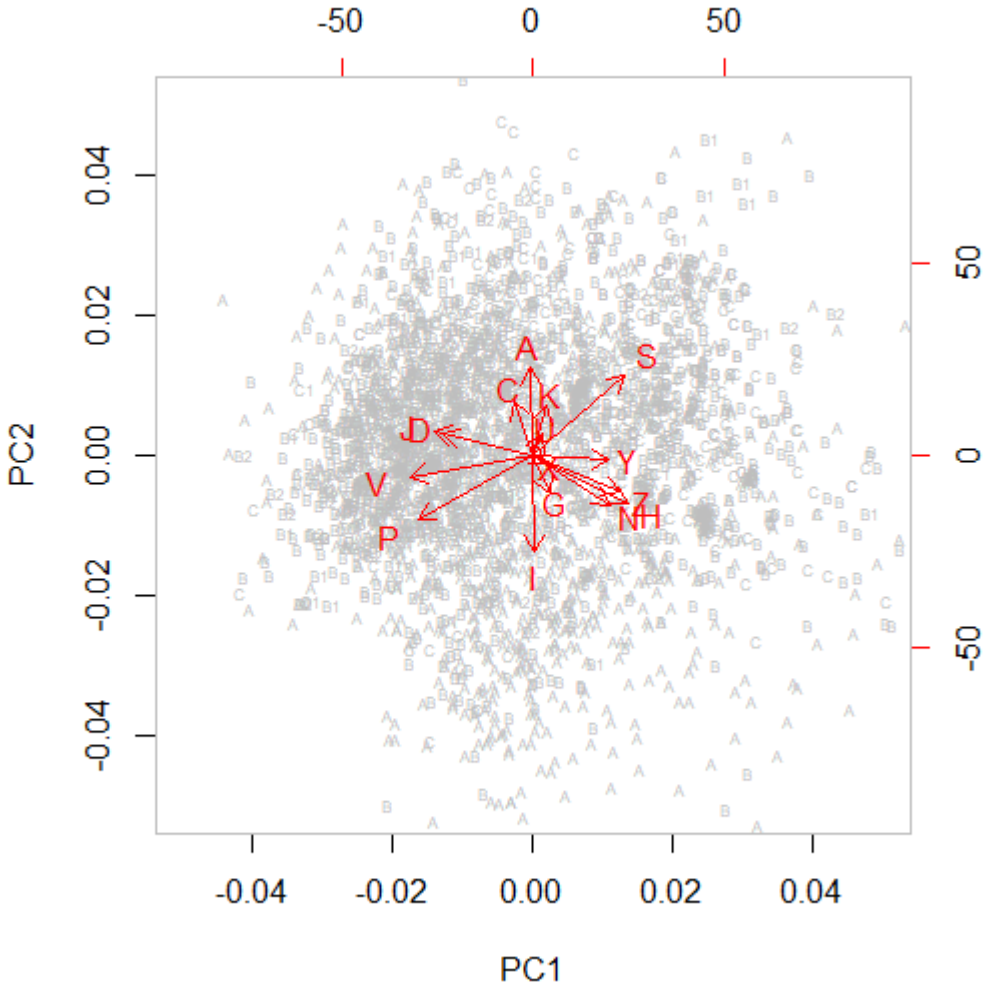
```
> koos  %>% select_if(is_numeric) %>% select(-kokku) %>% prcomp() %>%
biplot(xlabs=koos$keeletase, cex=c(0.5, 1), xlim=c(-0.05, 0.05), ylim=c(-0.05, 0.05),
expand=0.5, col=c("gray", "red"))
```

Endiselt suhteliselt kirju pilv, aga mõnda kohta tekivad siiski sama keeletasemega tekstide rühmad, mis annavad vihje nende sarnasuste põhjusi otsida.



Kui prcomp-käsule lisada parameeter scale=TRUE, siis kahaneb enne valitsevatena olnud nimisõnade, tegusõnade ning kirjavahemärkide ülemvõim ning ka vähem esindatud sõnaliigid pääsevad selgemalt esile

```
koos  %>% select_if(is_numeric) %>% select(-kokku) %>% prcomp(scale=TRUE) %>%
biplot(xlabs=koos$keeletase, cex=c(0.5, 1), xlim=c(-0.05, 0.05), ylim=c(-0.05, 0.05),
expand=0.5, col=c("gray", "red"))
```



Arvestades, et A ja B taseme tekste on pea võrdselt, hakkab A-taseme ülekaal äärealadel endiselt silma

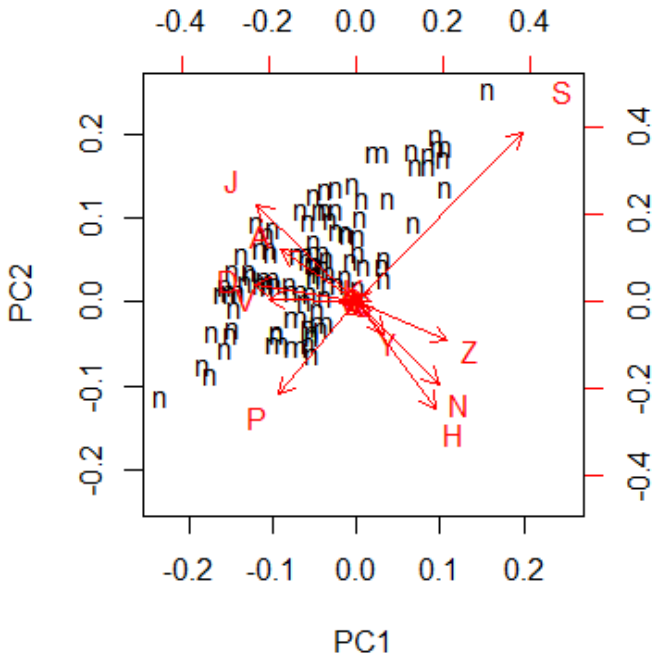
```
> koos %>% group_by(keeletase) %>% summarise(kogus=n()) %>% arrange(-kogus)
# A tibble: 9 × 2
  keeletase kogus
      <chr> <int>
1         B  1060
2         A  1051
3         C   353
4        B1   185
5        B2    84
6        A2    47
7        C1    42
8        A1     1
9        C2     1
```

Harjutus

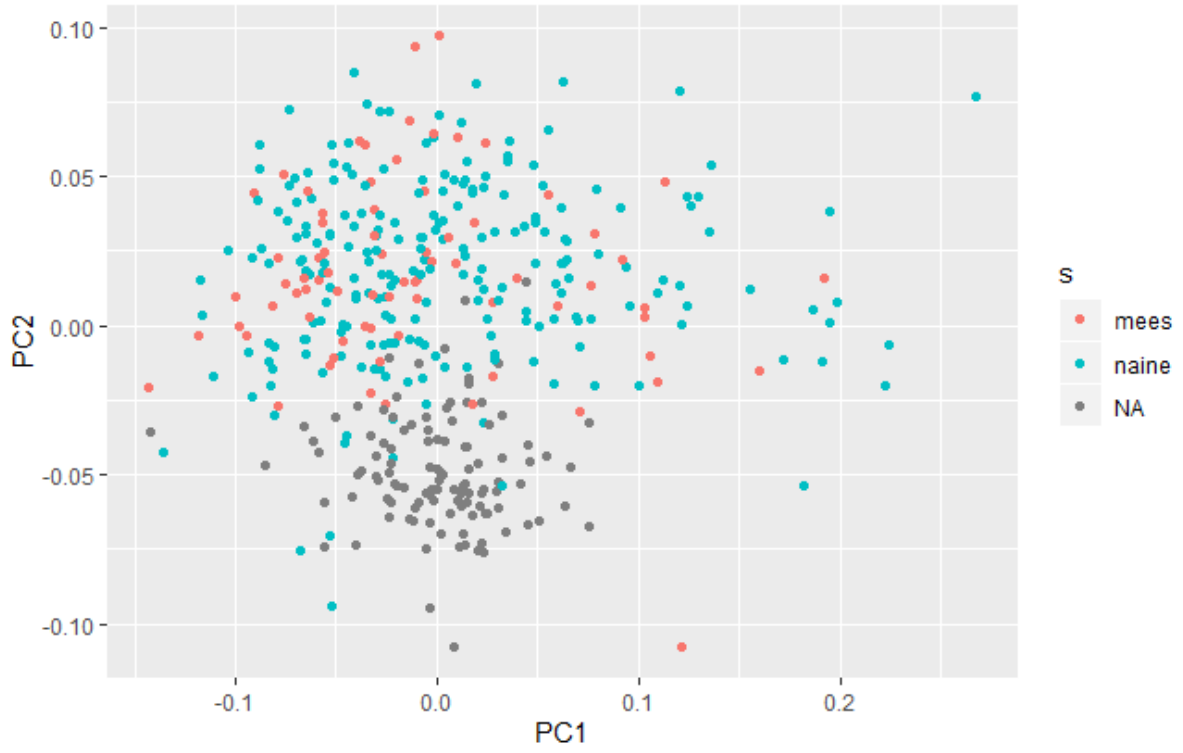
- Tehke näited läbi

- Kasutage ainult eestikeelseid A2-taseme tekste, jagage sõnaliikide arvud teksti pikkusega läbi. Näidake välja, kuidas tekstid peakomponentide biplot-joonisel paiknevad
- Tehke sarnane joonis B1-taseme tekstide kohta
- Kuvage B1-taseme tekstid kahe peakomponendi väärtuse järgi (prcomp-väljundi x-väärtus) värvige mehed ja naised eri värvi
- Koostage peakomponentide analüüs ilmaandmete näitel
  http://www.tlu.ee/~jaagup/andmed/ilm/harkutund.txt
  http://www.tlu.ee/~jaagup/andmed/ilm/selgitused.txt

```
doksonaliigid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt")
dokmeta=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt")
koos=dokmeta %>% inner_join(doksonaliigid) %>%
  mutate_if(is.numeric, funs(./doksonaliigid$kokku)) %>%
  filter(tekstikeel=="eesti") %>% filter(kokku>0) %>%
  select(-kokku)

a2tekstid=koos %>% filter(keeletase=="A2")
a2tekstid %>% select_if(is_numeric) %>%
  prcomp() %>% biplot(xlabs=substring(a2tekstid$sugu, 1, 1))
```

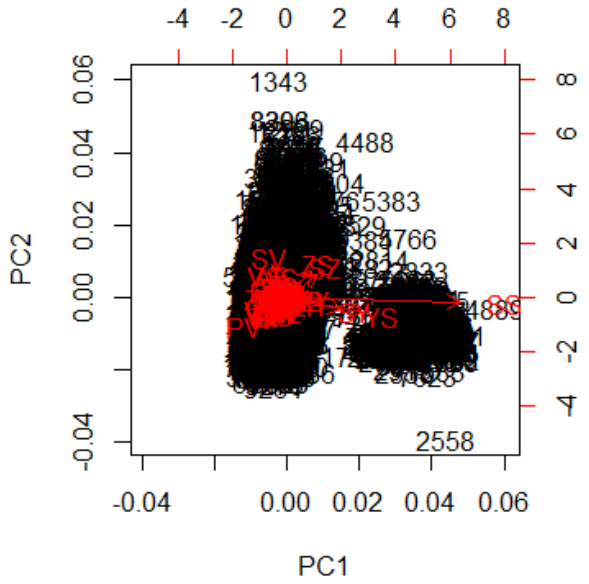



```
b1tekstid=koos %>% filter(keeletase=="B1")
b1tekstid %>% select_if(is_numeric) %>%
  prcomp() %>% .$x %>% as_tibble() %>% add_column(s=b1tekstid$sugu) %>%
  ggplot(aes(PC1, PC2, color=s))+geom_point()
```



Katse ngramidega

```
paarid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/ngram2.txt")
pikk=paarid %>% group_by(kood, ngram2) %>% summarise(kogus=n()) %>% ungroup()
head(pikk)
lai=pikk %>% spread(ngram2, kogus, fill=0)
lai %>% select(-kood) %>% prcomp() %>% summary()
mudel=lai %>% select(-kood) %>% prcomp()
mudel$rotation[, 1]
mudel %>% biplot()
lai %>% mutate_if(is_numeric,
      funs(./lai %>% select(-kood) %>% rowSums())) %>%
      select(-kood) %>% prcomp() %>% biplot()
```



Faktoranalüüs

Peakomponentide analüüs kirjeldab objektide asukohad teljestikus lõpuks täpselt ära, kasutades lõpuks sama palju tunnuseid/komponente, kui on algses andmestikus. Vastavalt soovitud täpsusele saab andmete esitamisel piirduda ainult osaga neist komponentidest.

Faktoranalüüsi puhul öeldakse juba ette, et mitme faktoriga kirjeldamise juures piirdutakse ning algoritm püüab olemasolevad tunnused paigutada võimalikult hästi nii mitme faktori alla. Kõigepealt näide sõnade kohta

```
>sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Kolme arvulise tunnuse ning kahe soovitud faktori kohta öeldakse, et kolme tunnust pole mõtete kahe faktori abil kirjeldada, faktoranalüüs on mõeldud tunnuste suuremaks kokku tõmbamiseks

```
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% factanal(factors=2)
Error in factanal(., factors = 2) :
  2 factors are too many for 3 variables
```

Ühe faktori puhul soovitatakse sõna iseloomustamiseks tunnust, mil on sõnapikkuse koefitsient 0,99, täishäälikute oma 0,9 ning sulghäälikute oma 0,7.

```
> sonad %>% select(sonapikkus, taishaalikuid, sulghaalikuid) %>% factanal(factors=1)

Call:
factanal(x = ., factors = 1)

Uniquenesses:
   sonapikkus taishaalikuid sulghaalikuid
```

```
        0.005         0.184         0.505

Loadings:
             Factor1
sonapikkus    0.998
taishaalikuid 0.903
sulghaalikuid 0.704
```

Järgmise näite juures on tunnuseid märgatavalt rohkem - iga teksti puhul seal leidunud sõnaliikide arv

```
> doksonaliigid=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/doksonaliigid.txt")
```

```
> head(doksonaliigid)
# A tibble: 6 x 18
  kood                    A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z kokku
  <chr>               <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 doc_100636852915_item    25     0    14     0     3     0    19     5     3    17    54     0    35     0     0    36   211
2 doc_100636852916_item     4     0     5     0     4     0    12     1     3    14    31     0    22     0     0    21   117
3 doc_100636852917_item     9     0     6     0     2     0    13     1     3    17    53     0    25     0     2    27   158
4 doc_1010138197_item      46     7    50     4    20     0    38     3     2    34   183     0   126     0     2   184   699
5 doc_1010138198_item      43     7    49     4    21     0    37     6     2    39   182     0   129     0     2   177   698
6 doc_1010138199_item      45     7    51     4    20     0    38     4     2    37   180     1   132     0     2   185   708
```

Leiame arvulised tulbad, jagame teksti pikkusega läbi, eemaldame puuduvate või nulliliste väärtustega read

```
> doksonaliigid %>% select_if(is_numeric) %>% {./.$kokku} %>% na.omit() %>% filter(kokku>0)
%>% select(-kokku) %>% round(2) %>% head()
     A    C    D    G    H I    J    K    N    P    S U    V X    Y    Z
1 0.12 0.00 0.07 0.00 0.01 0 0.09 0.02 0.01 0.08 0.26 0 0.17 0 0.00 0.17
2 0.03 0.00 0.04 0.00 0.03 0 0.10 0.01 0.03 0.12 0.26 0 0.19 0 0.00 0.18
3 0.06 0.00 0.04 0.00 0.01 0 0.08 0.01 0.02 0.11 0.34 0 0.16 0 0.01 0.17
4 0.07 0.01 0.07 0.01 0.03 0 0.05 0.00 0.00 0.05 0.26 0 0.18 0 0.00 0.26
5 0.06 0.01 0.07 0.01 0.03 0 0.05 0.01 0.00 0.06 0.26 0 0.18 0 0.00 0.25
6 0.06 0.01 0.07 0.01 0.03 0 0.05 0.01 0.00 0.05 0.25 0 0.19 0 0.00 0.26
```

Küsime, mida soovitab faktoranalüüs kolme faktoriga

```
> doksonaliigid %>% select_if(is_numeric) %>% {./.$kokku} %>% na.omit() %>% filter(kokku>0)
%>% select(-kokku) %>% factanal(factors=3)

Call:
factanal(x = ., factors = 3)

Uniquenesses:
    A     C     D     G     H     I     J     K     N     P     S     U     V     X     Y     Z
1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.437 1.000 0.789 1.000 0.939 1.000

Loadings:
  Factor1 Factor2 Factor3
A  0.320   0.345   0.217
C          0.338   0.274
D  0.592
G                  0.310
H -0.182  -0.441  -0.246
I  0.200  -0.249  -0.235
J  0.446   0.376   0.320
K          0.381
N  0.174  -0.187  -0.429
P  0.681
```

```
S -0.859
U          0.130   0.131
V  0.796   0.141
X
Y -0.736  -0.140
Z  0.134  -0.258  -0.257

               Factor1 Factor2 Factor3
SS loadings      3.153   0.937   0.725
Proportion Var   0.197   0.059   0.045
Cumulative Var   0.197   0.256   0.301

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 399236.5 on 75 degrees of freedom.
The p-value is 0
```

Paistab, et kolme faktori peale kokku suudetakse praegusel juhul kirjeldada vaid 0.301 ehk 30% andmete variatiivsusest ehk asukohast ruumis. Ühtlasi antakse laadungite (loadings) juures soovitused, et millise koefitsiendiga millist tunnust millise faktori juures arvestada. Esimesse faktorisse paistab suurimana tulema nimisõnade osakaal (S -0.859), samuti märgatavalt lühendid (Y -0.736), mis sageli nimisõnadega koos käivad ning tegusõnad (V 0.796). Teine ja kolmas faktor paistavad olema muude tunnuste järgi mõnevõrra ära jaotunud, aga selget kasulikku faktoriteks jagamist praeguste andmete juures ei paista. Vahel saab mängida faktoranalüüsi algoritmidega - eelistatakse tulemust, kus igale faktorile laadub selgelt paar tunnust - nii on lootusrikkam analüüsi tulemusi arusaadavalt tõlgendada. Seekord aga piisab teadmisest, et faktoranalüüs nende andmete puhul ei paista kuigi kasulik olema.

Mitmemõõtmeline skaleerimine (MDS)

Meetod paigutab objektid kaardile nende vahekauguste järgi. Lihtsaim näide on Eesti kaardiga, kuid hiljem vaatame ka meetodi ülekandmise võimalusi muu valdkonna andmetele

```
> vahemaad=read_csv("http://www.tlu.ee/~jaagup/andmed/muu/linnadevahemaad.txt")

> vahemaad
# A tibble: 11 × 12
    linnanimi  Elva Haapsalu Kuressaare Narva Pärnu Rakvere Tallinn Tartu Valga Viljandi
Võru
        <chr> <int>    <int>      <int> <int> <int>   <int>   <int> <int> <int>    <int>
<int>
1        Elva     0      267        308   211   156     152     216    27    60       70
75
2    Haapsalu   267        0        155   314   111     203     101   258   254      199
310
3  Kuressaare   308      155          0   429   152     315     216   330   295      249
351
4       Narva   211      314        429     0   299     116     212   184   271      265
252
5       Pärnu   156      111        152   299     0     183     129   178   143       97
199
```

```
6     Rakvere   152      203        313   116   183       0      99   126   212      151
193
7     Tallinn   216      101        216   212   129      99       0   189   252      161
257
8       Tartu    27      258        330   184   178     126     189     0    87       73
68
9       Valga    60      254        295   271   143     212     252    87     0       91
71
10   Viljandi    70      199        249   265    97     151     161    73    91        0
128
11       Võru    75      310        351   252   199     193     257    68    71      128
0
```

Vahemaad ühekordsena

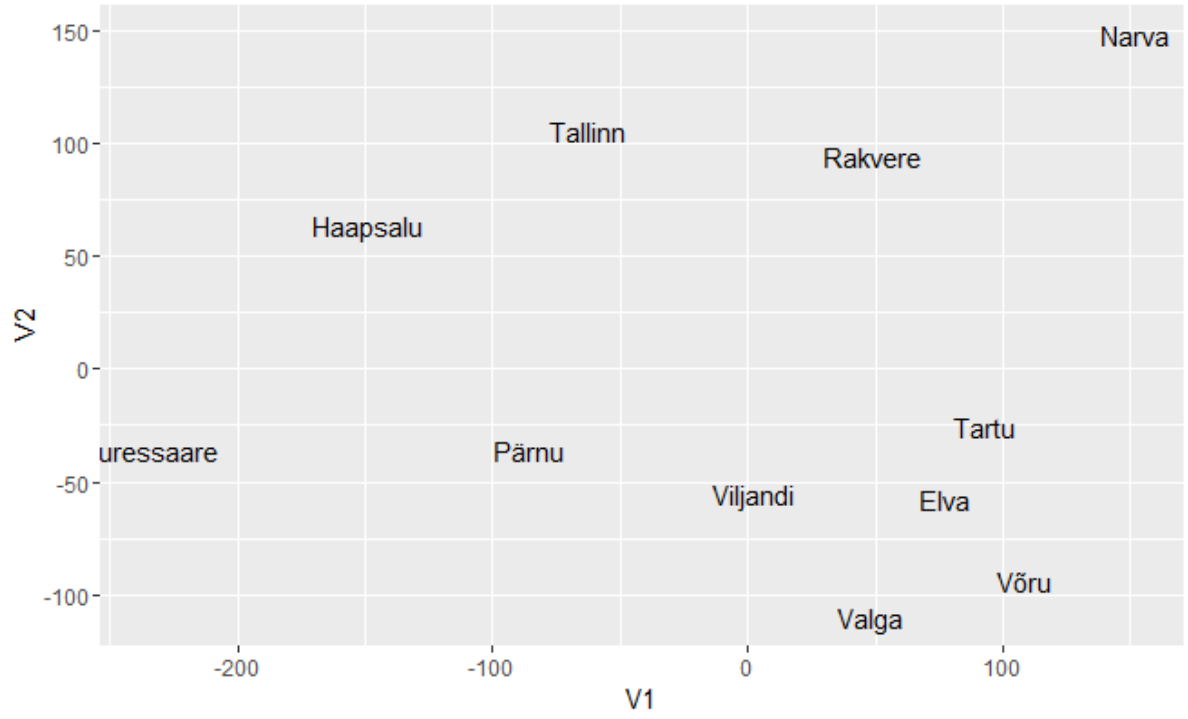
```
> vahemaad %>% select(-linnanimi) %>% as.dist()
           Elva Haapsalu Kuressaare Narva Pärnu Rakvere Tallinn Tartu Valga Viljandi
Haapsalu    267
Kuressaare  308      155
Narva       211      314        429
Pärnu       156      111        152   299
Rakvere     152      203        313   116   183
Tallinn     216      101        216   212   129      99
Tartu        27      258        330   184   178     126     189
Valga        60      254        295   271   143     212     252    87
Viljandi     70      199        249   265    97     151     161    73    91
Võru         75      310        351   252   199     193     257    68    71      128
```

Skaleerimise abil kahemõõtmelisele tasandile

```
> vahemaad %>% select(-linnanimi) %>% as.dist() %>% cmdscale(2)
                  [,1]       [,2]
Elva         77.094325  -57.43633
Haapsalu   -148.202384   63.85243
Kuressaare -234.229463  -35.61484
Narva       151.873137  148.46268
Pärnu       -85.071000  -35.56523
Rakvere      48.665910   94.39486
Tallinn     -62.567317  105.87773
Tartu        92.678256  -25.60180
Valga        48.034540 -109.71883
Viljandi      3.070653  -54.94348
Võru        108.653344  -93.70718
```

Koos linnanimedega kaardile

```
> vahemaad %>% select(-linnanimi) %>% as.dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(linnanimi=vahemaad$linnanimi) %>% ggplot(aes(V1, V2,
label=linnanimi))+geom_text()
```



Välja paistab suhteliselt tuttav kaart - meetod nihutas linnad tasandile nõnda, et nende vahekaugused jäid võimalikult sarnaselt võrdelisteks sisendtabelis olevate kaugustega

Meetodi ülesehituse tutvustamiseks määrame Tallinna ja Haapsalu vahemaa "ümbersõidu tõttu" pikemaks, et selgelt näha oleks, siis kolm korda pikemaks

```
> vahemaad[2, "Tallinn"] <- vahemaad[7, "Haapsalu"] <- 300
```

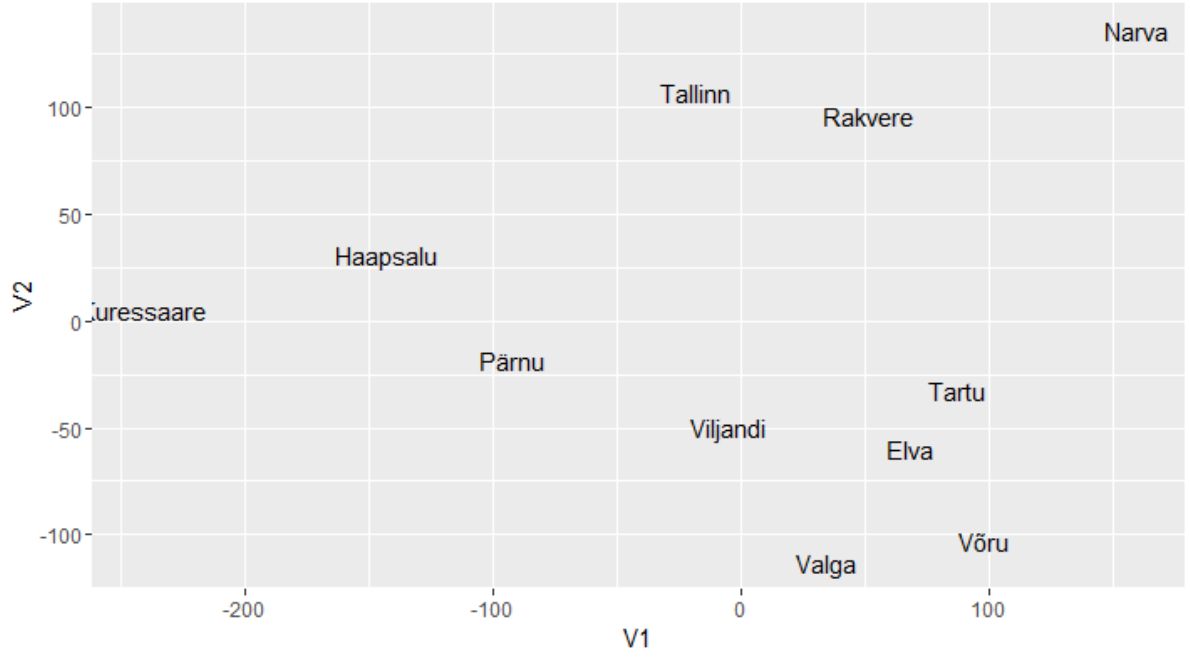
See arv kahes kohas suurem, muud väärtused samasugused

```
> vahemaad
# A tibble: 11 × 12
    linnanimi  Elva Haapsalu Kuressaare Narva Pärnu Rakvere Tallinn Tartu Valga Viljandi
Võru
        <chr> <int>    <dbl>      <int> <int> <int>   <int>   <dbl> <int> <int>    <int>
<int>
1        Elva     0      267        308   211   156     152     216    27    60       70
75
2    Haapsalu   267        0        155   314   111     203     300   258   254      199
310
3  Kuressaare   308      155          0   429   152     315     216   330   295      249
351
4       Narva   211      314        429     0   299     116     212   184   271      265
252
5       Pärnu   156      111        152   299     0     183     129   178   143       97
199
6     Rakvere   152      203        313   116   183       0      99   126   212      151
193
7     Tallinn   216      300        216   212   129      99       0   189   252      161
257
8       Tartu    27      258        330   184   178     126     189     0    87       73
68
```

```
9       Valga    60      254        295   271   143     212     252    87     0        91
71
10   Viljandi    70      199        249   265    97     151     161    73    91         0
128
11       Võru    75      310        351   252   199     193     257    68    71       128
0
```

Kui nüüd kaart joonistada, siis on Tallinn ja Haapsalu teineteisest lahku nihutatud. Mitte küll niipalju nagu nende omavaheline määratud kilomeetrite arv näidanuks, aga nähtavalt siiski. Kaugused muude linnadega tasandavad seda ühte märgatavat erinevust

```
> vahemaad %>% select(-linnanimi) %>% as.dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(linnanimi=vahemaad$linnanimi) %>% ggplot(aes(V1, V2,
label=linnanimi))+geom_text()
>
```



Näide sõnadega

Tuttavate sõnadega

```
> sonad %>% head(5)
# A tibble: 5 x 5
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid
  <chr>  <chr>        <int>         <int>         <int>
1 kungla kui              3             2             1
2 kungla kungla           6             2             2
3 kungla rahvas           6             2             0
4 kungla kuldsel          7             2             2
5 kungla aal              3             2             0
```

Sõnade vahelised kaugused suhtelistes ühikutes

```
> sonad %>% head(5) %>% select_if(is_numeric) %>% dist()
         1        2        3        4
2 3.162278
3 3.162278 2.000000
4 4.123106 1.000000 2.236068
5 1.000000 3.605551 3.000000 4.472136
```

Algoritme mitmesuguseid - üheks mooduseks on võtta erinevus suurima erinevusega tunnuse järgi. Näiteks sõnade "kungla" ja "rahvas" vahel on erinevus 2, sest esimesel on kaks sulghäälikut ja teisel pole ühtegi

```
> sonad %>% head(5) %>% select_if(is_numeric) %>% dist(method="maximum")
  1 2 3 4
2 3
3 3 2
4 4 1 2
5 1 3 3 4
```

Meetodi abil asukohad koordinaatteljestikul

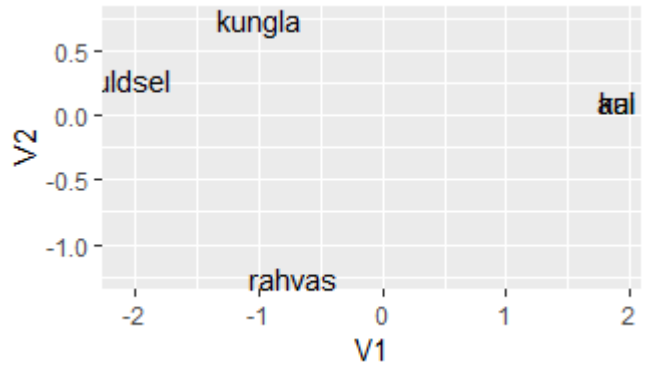
```
> sonad %>% head(5) %>% select_if(is_numeric) %>% dist(method="maximum") %>% cmdscale(2)
           [,1]       [,2]
[1,]  1.9027986  0.1118456
[2,] -1.0087115  0.7409124
[3,] -0.7283762 -1.2488042
[4,] -2.0685094  0.2842006
[5,]  1.9027986  0.1118456
```

Juurde sõnad

```
> sonad %>% head(5) %>% select_if(is_numeric) %>% dist(method="maximum") %>% cmdscale(2)
%>% as_tibble() %>% add_column(sonasisu=sonad$sona[1:5])
# A tibble: 5 x 3
      V1     V2 sonasisu
   <dbl>  <dbl> <chr>
1  1.90   0.112 kui
2 -1.01   0.741 kungla
3 -0.728 -1.25  rahvas
4 -2.07   0.284 kuldsel
5  1.90   0.112 aal
```

ning kogu ettevõtmine joonisena

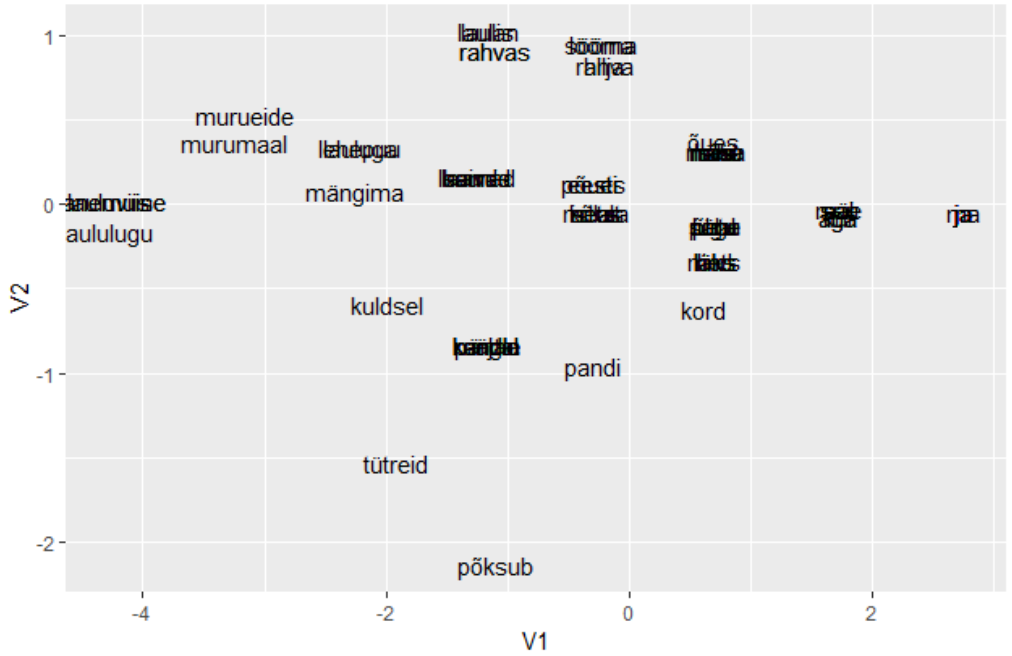
```
> sonad %>% head(5) %>% select_if(is_numeric) %>% dist(method="maximum") %>% cmdscale(2)
%>% as_tibble() %>% add_column(sonasisu=sonad$sona[1:5]) %>% ggplot(aes(V1, V2,
label=sonasisu)) + geom_text()
```



Harjutus

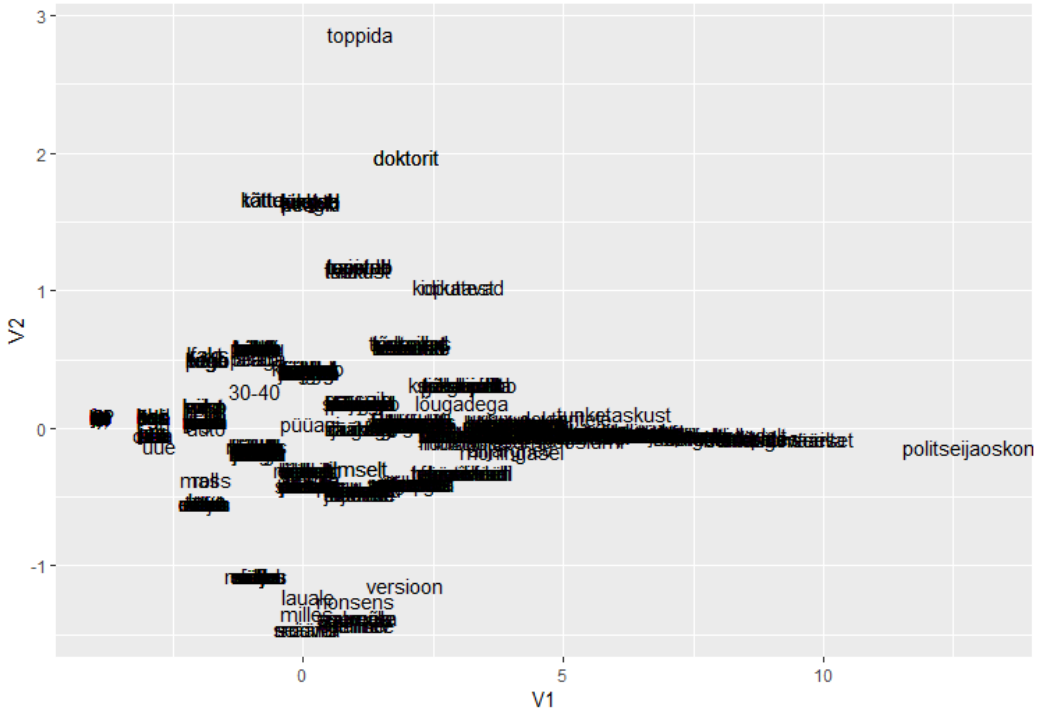
- Koostage sarnane kahemõõtmeline MDS joonis kogu Kungla rahva laulu sõnade kohta

```
> kunglarahvas <- sonad %>% filter(lugu=="kungla")

> kunglarahvas %>% select_if(is_numeric) %>% dist(method="maximum") %>% cmdscale(2) %>%
as_tibble() %>% add_column(sonasisu=kunglarahvas$sona) %>% ggplot(aes(V1, V2,
label=sonasisu)) + geom_text()
```



Juurde näide, kuidas dplyri käskudeahelas saab vahetulemuse meelde jätta, et seda hiljem kasutada. Ahelas olev `{. ->> andmed}` jätab jooksva seisu muutujasse andmed ning sealt saab selle välja võtta, et MDSi abil välja arvutatud koordinaatidele sõnad külge panna. Nii on võimalik ühe filter-käsuga määrata, milliseid andmeid analüüsis kasutatakse ning hiljem sama valiku juurde jääda vastuste välja näitamise juures.

```
> sonad %>% filter(lugu=="lambipirn") %>% {. ->> andmed} %>% select_if(is_numeric) %>%
dist(method="maximum") %>% cmdscale(2) %>% as_tibble() %>% add_column(sonasisu=andmed$sona)
%>% ggplot(aes(V1, V2, label=sonasisu)) + geom_text()
```



Tähtede sagedused eesti ja soome keeles

Mitmemõõtmelist skaleerimist kasutatakse pigem olukordades, kus tunnuseid on palju ja sageli märgatavalt erinevate väärtustega. Faili teksti lugemiseks sobib käsklus read_file - olgu siis kohalikust kataloogist või võrgust

```
> read_file("http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt")
[1] "Kavastust mõni kilomeeter eemale Kantsi kõrtsi varemetele rajatud Emajõe Suursoo
looduskeskuse uhkest hoonest pole seni õnnestunud elavalt kasutatavat inimese ja looduse
kohtumiskohta teha, maja ei ole huvilistele pidevalt avatud. Nüüd tahab sellele keskusele
elu sisse puhuda Luunja vald, soovides hoone ja selle ümbruse ka tasuta
võõrandada.\r\n\r\nEmajõe Suursoo looduskeskus kuulub riigimetsa majandamise keskusele
(RMK), mis 2016. aasta juulist loobus seal looduskeskuse pidamisest.
```

Lihtsama uurimise huvides saab tähed väikeseks teha

```
> failinimi<- "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt"
> tekst <- read_file(failinimi)
> str_to_lower(tekst)
[1] "kavastust mõni kilomeeter eemale kantsi kõrtsi varemetele rajatud emajõe suursoo
looduskeskuse uhkest hoonest pole seni õnnestunud elavalt kasutatavat inimese ja looduse
kohtumiskohta teha,
```

Tühja teksti kohalt tükeldades saab kõik tähed eraldi kätte

```
> str_split(str_to_lower(tekst), "")
[[1]]
```

```
   [1] "k"    "a"    "v"    "a"    "s"    "t"    "u"    "s"    "t"    " "
  [11] "m"    "õ"    "n"    "i"
```

Et vastus oli listina, tuleb sealt "puhaste" tähtede saamiseks veel esimene element küsida

```
> str_split(str_to_lower(tekst), "")[[1]]
   [1] "k"    "a"    "v"    "a"    "s"    "t"    "u"    "s"    "t"    " "
  [11] "m"    "õ"    "n"    "i"
```

Vormistame failist tulevate tähtede sagedused eraldi funktsioonina

```
tahtedeSagedused <- function(failinimi){
  tekst= read_file(failinimi)
  vastus=tibble(taht=str_split(str_to_lower(tekst), "")[[1]]) %>% group_by(taht) %>%
    summarise(kogus=n())
  return (vastus)
}
```

Postimehe artiklist leitud tähed sageduse järjekorras

```
tahtedeSagedused("http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt") %>%
arrange(-kogus)

# A tibble: 40 x 2
   taht  kogus
   <chr> <int>
 1 " "     405
 2 a       363
 3 e       289
 4 s       275
 5 i       183
 6 u       181
 7 l       177
 8 t       163
 9 k       123
10 n       114
# ... with 30 more rows
>
```

Mitme artikli tähtede sagedused

```
kataloog="http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/"
failinimed=c("hsanomat1.txt", "hsanomat2.txt", "postimees1.txt", "postimees2.txt")
```

Käsklus paste aitab tekstid üheks liita. Parameeter sep="" (tühi tekst) näitab, et tagumine osa tuleb kohe esimese otsa panna.

```
> paste(kataloog, failinimed[1], sep="")
[1] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat1.txt"
```

Käsu full_join puhul jäetakse alles mõelmas seotud tabelis olevad tähed. Hilisem

```
koos=koos %>% replace(., is.na(.), 0)
```

hoolitseb, et tühjaks jäänud kohtadesse pannakse nullid - ehk siis vastavas tekstis vastavat sümbolit ei olnud.

```
koos=tahtedeSagedused(paste(kataloog, failinimed[1], sep=""))
colnames(koos)=c("taht", failinimed[1])
for(failinimi in failinimed[2:length(failinimed)]){
  tabel=tahtedeSagedused(paste(kataloog, failinimi, sep=""))
  colnames(tabel)=c("taht", failinimi)
  koos=koos %>% full_join(tabel, by="taht")
}
koos=koos %>% replace(., is.na(.), 0)
print(koos %>% arrange(-postimees1.txt))
```

Võrdlev tabel tähtede sageduse kohta neljas tekstis

```
> print(koos %>% arrange(-postimees1.txt))
# A tibble: 51 x 5
   taht  hsanomat1.txt hsanomat2.txt postimees1.txt postimees2.txt
   <chr>         <dbl>         <dbl>          <dbl>          <dbl>
 1 " "             134           176            405            380
 2 a               109           143            363            281
 3 e                89            90            289            204
 4 s                76            97            275            207
 5 i               106           156            183            241
 6 u                45            44            181            164
 7 l                44            80            177            148
 8 t               101           123            163            195
```

Read ja veerud pööratuna, tähed ise välja, et jääksid arvulised andmed alles

```
> koos %>% select(-taht) %>% t()
               [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
hsanomat1.txt     6    1  134   10    9   13    3    4    3     3     3
hsanomat2.txt     0    1  176   12   17   14    0    4    0     0     0
postimees1.txt    3    1  405   22   33   35    0    0    3     2     0
postimees2.txt    5    0  380   23   36   26    2    0    6     1     3
```

Tekstide asukohad koordinaatteljestikul

```
> koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2)
                    [,1]      [,2]
hsanomat1.txt  -260.7165 -31.58916
hsanomat2.txt  -192.6546  17.81717
postimees1.txt  280.3549 -66.94723
postimees2.txt  173.0162  80.71922
```

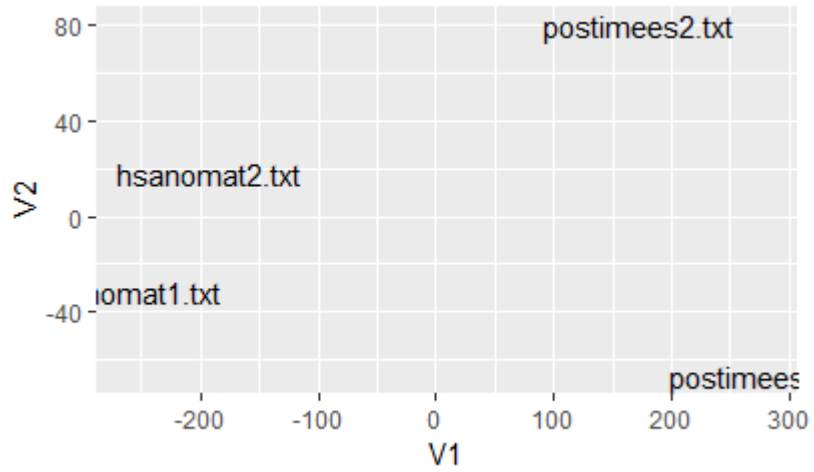
Failinimed omaette tulbaks - kust joonise koostamisel saab nad kergemini kätte

```
> koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=failinimed)
# A tibble: 4 x 3
     V1    V2 failinimi
  <dbl> <dbl> <chr>
1 -261. -31.6 hsanomat1.txt
```

```
2 -193.  17.8 hsanomat2.txt
3  280. -66.9 postimees1.txt
4  173.  80.7 postimees2.txt
```

Tekstid joonisele

```
> koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=failinimed) %>% ggplot(aes(V1, V2, label=failinimi)) + geom_text()
```



Nagu näha, siis soome tekstid on mõnevõrra rohkem omavahel koos, aga kaks teksti kummastki keelest on suuremate järelduste tegemiseks veel vähevõitu.

Võrdlus markertekstidega

Neli ajaleheteksti jätame võrdluseks ehk markeriks, mille järgi saab vaadata, kuhu lisanduvad andmed paiknevad. Uuritavateks tekstideks võtame kättesaadavad eesti keele õppijate tekstid koos metaandmetega

```
> dokmeta=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt")

> head(dokmeta)
# A tibble: 6 x 13
  kood                  korpus    tekstikeel tekstityyp elukoht taust vanus  sugu  emakeel kodukeel keeletase haridus
abivahendid
  <chr>                 <chr>     <chr>      <chr>      <chr>   <chr> <chr>  <chr> <chr>   <chr>    <chr>     <chr>   <chr>
1 doc_100636852915_item cFOoRQekA eesti      essee      idaviru op    kuni18 naine vene    vene     B         pohi    ei
2 doc_100636852916_item cFOoRQekA eesti      muu        idaviru op    kuni18 naine vene    vene     B         pohi    ei
3 doc_100636852917_item cFOoRQekA eesti      essee      idaviru op    kuni18 naine vene    vene     B         pohi    ei
4 doc_1010138197_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     A         kesk    ei
5 doc_1010138198_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     B         kesk    ei
6 doc_1010138199_item   cFOoRQekA eesti      muu        tallinn ylop  kuni26 naine vene    vene     A         kesk    ei
>
```

Neist valime tekstid, kus autori emakeel on soome keel

```
> dokmeta %>% filter(emakeel=="soome")
# A tibble: 391 x 13
```

```
   kood                korpus    tekstikeel tekstityyp elukoht taust   vanus  sugu  emakeel kodukeel keeletase haridus
abivahendid
   <chr>               <chr>     <chr>      <chr>      <chr>   <chr>   <chr>  <chr> <chr>   <chr>    <chr>     <chr>
<chr>
 1 doc_104580264060_item cFOoRQekA eesti      muu        soome   teenist kuni40 naine soome   soome    B1        kesk    jah
 2 doc_104580264061_item cFOoRQekA eesti      essee      soome   teenist kuni40 naine soome   soome    C1        kesk    jah
 3 doc_104580264062_item cFOoRQekA eesti      essee      soome   teenist kuni40 naine soome   soome    B2        kesk    jah
 4 doc_104580264063_item cFOoRQekA eesti      essee      soome   teenist kuni26 naine soome   soome    NA        kesk    jah
 5 doc_104580264064_item cFOoRQekA eesti      essee      soome   teenist kuni26 naine soome   soome    C1        kesk    jah
 6 doc_104580264065_item cFOoRQekA eesti      harjutus   soome   teenist kuni26 naine soome   soome    B1        kesk    jah
 7 doc_104580264066_item cFOoRQekA eesti      muu        soome   teenist kuni40 naine soome   soome    B1        kesk    jah
 8 doc_104580264067_item cFOoRQekA eesti      muu        soome   teenist kuni26 naine soome   soome    B1        kesk    jah
 9 doc_104580264068_item cFOoRQekA eesti      muu        soome   teenist kuni26 naine soome   soome    B1        kesk    jah
10 doc_104580264069_item cFOoRQekA eesti      batoo      soome   teenist kuni26 naine soome   soome    A2        kesk    jah
```

Kontrollime igaks juhuks üle, et soome emakeelega õppijatelt on vaid eestikeelsed tekstid - andmestikus endas leidub ka venekeelseid tekste

```
> dokmeta %>% filter(emakeel=="soome") %>% .$tekstikeel %>% unique()
[1] "eesti"
```

Tekstide leidmiseks on vaja kätte saada teksti kood

```
> dokmeta %>% filter(emakeel=="soome") %>% .$kood
  [1] "doc_104580264060_item" "doc_104580264061_item" "doc_104580264062_item"
  [4] "doc_104580264063_item" "doc_104580264064_item" "doc_104580264065_item"
```

Koodi järgi saab avada teksti vastavas kataloogis

http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264060_item.txt

```
 3-5 kartulid  2-3 porgandid  1 väike moorputk  1 sibul  küüslaugu  1 väike lillkapsas  5
dl keedud ubi  2 rkl oliiviõli  currytahna  köömneseemni     Koori ja tükelda sibuli ja
küüslaugut ja praadi neid köömneseemnedega oliiviõlis katlas.
Lisa currytahna katlasse.
Koori ja tükelda kartulid, porgandid ja moorputk.
Lisa ned katlasse ja hauta umbes 10 minutit kaane all.
Tükelda lillkapsas ja lisa see katlasse, kui kartulid on pooliks küpsaks saanud.
Lisa tilk vett, kui on vajalik.
Kui lillkapsas ja juureviljad on küpsad, lisa keedud ubad.
Hauta veel hetk.

Serveeri basmatiriisi ja naanleibaga.
```

Tekstide andmete lugemiseks tuleb vastavad aadressid kättesaadaval kujul kokku panna. Kõigepealt ajalehetekstide omad

```
kataloog="http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/"
failinimed=c("hsanomat1.txt", "hsanomat2.txt", "postimees1.txt", "postimees2.txt")
asukohad=paste(kataloog, failinimed, sep="")
> asukohad
[1] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat1.txt"
[2] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat2.txt"
[3] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt"
[4] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees2.txt"
```

ja eraldi keeleõppijate omad

```
kataloog2="http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/"
failinimed2=dokmeta %>% filter(emakeel=="soome") %>% .$kood
asukohad2=paste(kataloog2, failinimed2, ".txt", sep="")

> head(asukohad2)
[1] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264060_item.txt"
[2] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264061_item.txt"
[3] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264062_item.txt"
```

Lõpuks saab need omavahel ühendada

```
> asukohad = c(asukohad, asukohad2)
> head(asukohad)
[1] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat1.txt"
[2] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat2.txt"
[3] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt"
[4] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees2.txt"
[5] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264060_item.txt"
[6] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264061_item.txt"
```

ja tekstide tähtede sagedused kokku küsida nii, et igas tulbas omaette teksti andmed, iga rida on ühe tähe tarbeks

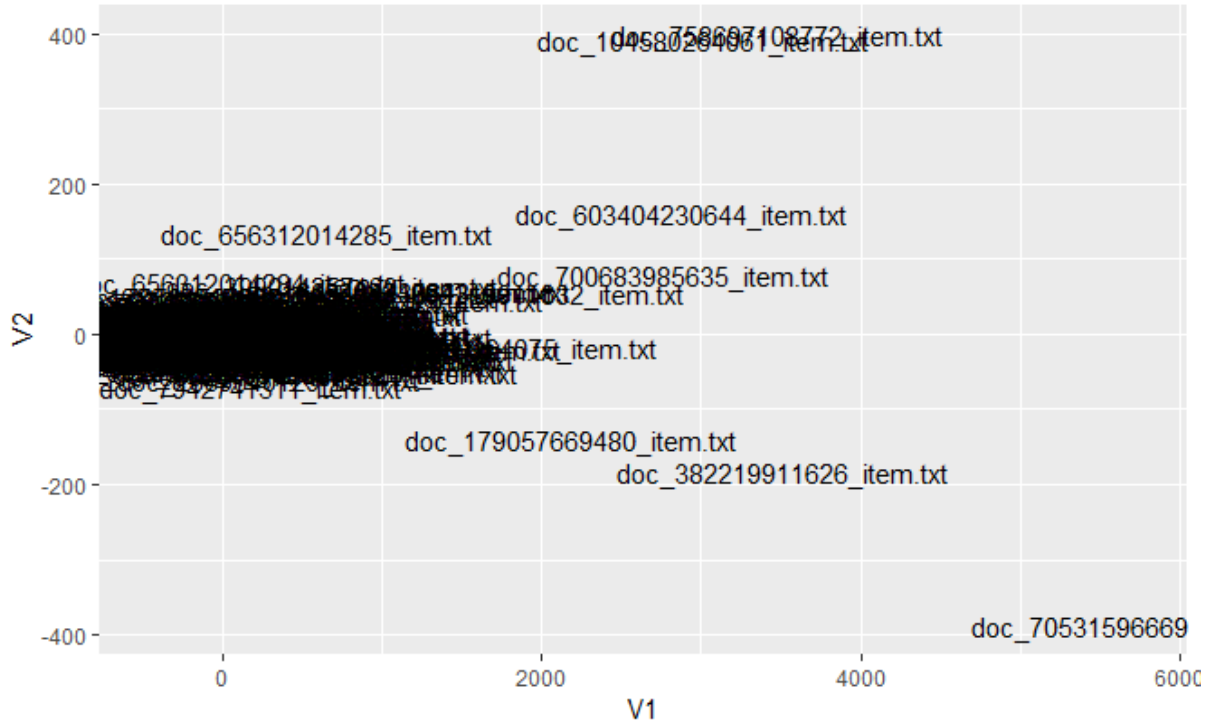
```
koos=tahtedeSagedused(asukohad[1])
colnames(koos)=c("taht", basename(asukohad[1]))
for(failinimi in asukohad[2:length(asukohad)]){
  tabel=tahtedeSagedused(failinimi)
  colnames(tabel)=c("taht", basename(failinimi))
  koos=koos %>% full_join(tabel, by="taht")
}
koos=koos %>% replace(., is.na(.), 0)
```

Skaleerimiskäsu abil saab igale tekstile koordinaadid tasandil

```
> koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=basename(asukohad))
# A tibble: 395 x 3
        V1       V2 failinimi
     <dbl>    <dbl> <chr>
 1  -172.    10.9   hsanomat1.txt
 2   -91.5    6.17  hsanomat2.txt
 3   340.   -45.6   postimees1.txt
 4   262.   -30.7   postimees2.txt
 5  -309.     0.858 doc_104580264060_item.txt
 6  3014.   393.    doc_104580264061_item.txt
 7   187.   -10.4   doc_104580264062_item.txt
 8   -53.3   18.2   doc_104580264063_item.txt
 9  1042.    57.7   doc_104580264064_item.txt
10  -279.    20.0   doc_104580264065_item.txt
# ... with 385 more rows
```

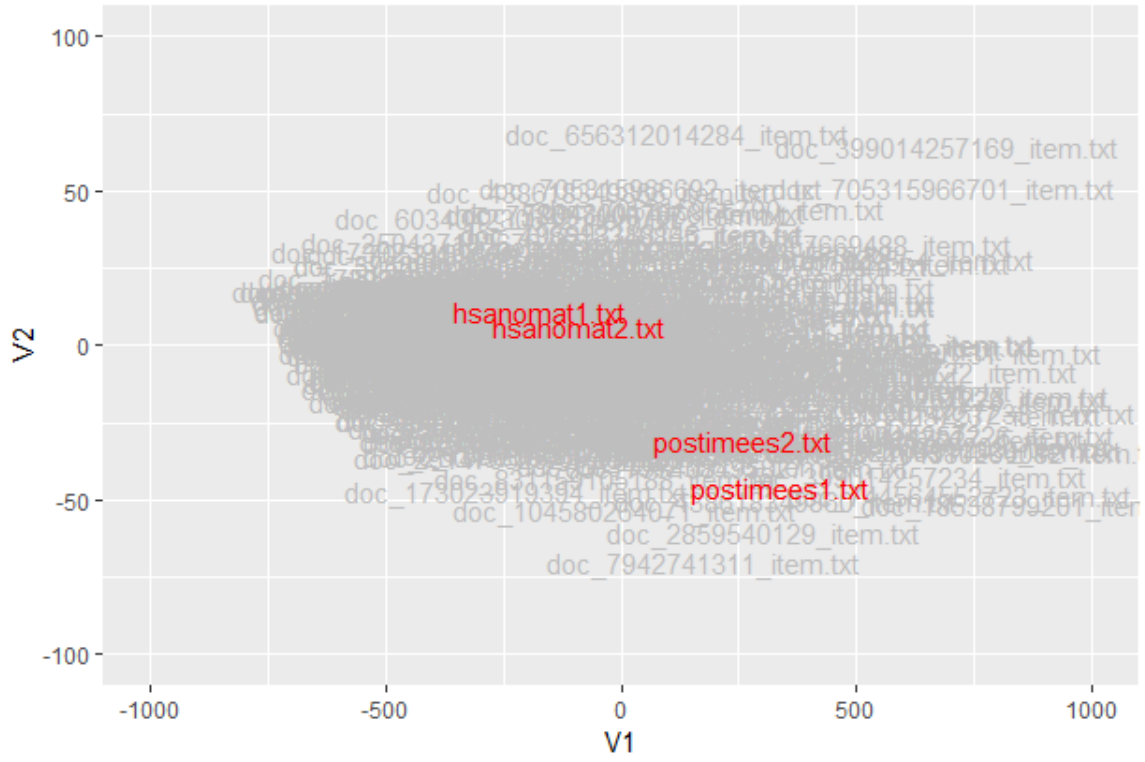
Nende abil saab askohad ekraanile joonistada

```
> koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=basename(asukohad)) %>% ggplot(aes(V1, V2, label=failinimi)) +
geom_text()
```



Andmeid aga nõnda palju, et ajalehetekstid ei paista teiste hulgast välja. Muudame esialgse joonise halliks ning lisame eraldi geom_text käsuga nimekirja neli esimest punasena juurde - siis on tulemus paremini nähtav. Vahepealne töödeldud andmete puhvrisse paigutamine võimaldab pärast samal kujul joonistamiseks mugavad andmed sealt jälle välja võtta. Joonistusala piiramine jätab mõned teistest kaugemale jäänud tekstid välja ning sisemist osa on mugavam lähemalt uurida.

```
koos %>% select(-taht) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=basename(asukohad)) %>% {. ->>puhver} %>% ggplot(aes(V1, V2,
label=failinimi)) + geom_text(color="gray") + geom_text(data=puhver %>% head(4),
color="red") + xlim(-1000, 1000) +ylim(-100, 100)
```



Faili postimees1.txt lähedal paistab olema tekst

http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_2859540129_item.txt

Kursuse alguses läksin raamatukokku ja validsin sealt huvitava romaani kursusele loetavaks.
Ma ei teadnud mitte midagi Eesti kirjandusest, aga sattumisi validsin ühe kõige tähtsama kirjaniku raamatu.
See kirjanik oli Mats Traat, kes on kirjutanud luuletusi ja romaane.
Mats Traat sündis Otepääs aastal 1936.

Joonise ülaservas soomekeelsete ajalehetekstide lähemal on

http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_656312014284_item.txt

Kakskeelne traditsioon pole Eestis pikk ja juurdunud kui see on Soomes, vaid eestikeelne ja venekeelne maailm on eraldi teine teisest.
Peale selle on emotsioonid tugevad poole ja vastu.
Minu arvamusel aitab ainult aeg selle.
Aga väga kiiresti peaks otsustada kuidas võidakse takistada venekeelse vähemuse kõrvale jäämine Eestis.

Silma järgi vaadates ei paista suuri keelelisi erinevusi, kuid mõni soome keelele lähedasem käändekasutus tundub ehk rohkem olema "peaks otsustada", "poole ja vastu".

Võrdlus rühmade kaupa

Üksikobjektide rohkus suudab joonise kergesti ummistada. Üldsuundade välja toomiseks sobib rühmade kaupa andmed keskmistada ning siis näeb juba rühmade asukohti joonisel (kuhu võivad ka üksikobjektid alles jääda)

Siinses näites püüame võrrelda soomlastest eesti keele õppijate tähekasutust keeletasemete kaupa. Kasutatud dokmeta-tabelis on mõnedel tekstidel kolmeastmeline keeletase (A - algaja, B-edasijõudnud õppija, C-emakeelekõneleja), mõnedel kuueastmeline (A1, A2, B1, B2, C1, C2) ning paljudel tekstidel puudub taseme määrang sootuks.

```
> dokmeta %>% select(kood, keeletase)
# A tibble: 12,724 × 2
                    kood keeletase
                   <chr>     <chr>
1  doc_100636852915_item         B
2  doc_100636852916_item         B
3  doc_100636852917_item         B
4    doc_1010138197_item         A
5    doc_1010138198_item         B
6    doc_1010138199_item         A
7    doc_1010138200_item         A
8  doc_101672866015_item         C
9  doc_104580264035_item      <NA>
10 doc_104580264036_item        C1
# ... with 12,714 more rows
```

Asukohtade loetelus on praeguste allikate veebiaadressid

```
> head(asukohad)
[1] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat1.txt"
[2] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/hsanomat2.txt"
[3] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt"
[4] "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees2.txt"
[5] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264060_item.txt"
[6] "http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/doc_104580264061_item.txt"
```

Tabelis dokmeta on kood kujul doc_104580264060_item - see tähendab, et tabelite ühendamiseks on kasulik eelnev kataloginimi ning järgnev .txt sealt maha võtta. Õnneks aitab funktsioon gsub, kus regulaaravaldise abil pääseb määrama, mis võtta ja mis alles jätta.

Avaldis .*/(.*).txt jagab teksti kujule, kus lõpus on .txt, algul on kaldkriipsuga lõppevad suvalised sümbolid ning sulgude sisse jäävad muud suvalised sümbolid. Kuna tavalise otsingu puhul võetakse algusest iga osa nii pikk kui võimalik samas, et teised ka täidetud saaksid, siis nõnda õnnestubki sobiv kood sulgude seest kätte saada. Hilisema asenduse juures viidatakse sellele kui vastusele \1, langjoone erisümboli staatuse tõttu "\\1".

```
> gsub(".*/(.*).txt", "\\1", asukohad)
  [1] "hsanomat1"             "hsanomat2"             "postimees1"            "postimees2"
```

```
  [5] "doc_104580264060_item" "doc_104580264061_item" "doc_104580264062_item"
"doc_104580264063_item"
  [9] "doc_104580264064_item" "doc_104580264065_item" "doc_104580264066_item"
"doc_104580264067_item"
 [13] "doc_104580264068_item" "doc_104580264069_item" "doc_104580264070_item"
"doc_104580264071_item"
```

Mugavama vaatamise huvides lisame asukohakoodide tulba esimeseks (.before=1 ehk enne praegust esimest tulpa) ja lisame sinna kõrvale left_join-i abil dokmeta-tabeli keeletaseme tulba. Vasakpoolne ühendamine seetõttu, et esialgse tabeli kõik read jääksid alles. Oleks ka võimalik ajalehtede andmeid omaette muutujas hoida ning õppijatekstide andmeid eraldi töödelda ning pärast alles ühte panna. Kui rühmi rohkem ja nad vajaksid erisugust töötlust, siis võib see isegi loetavama tulemuse anda. Et ka keeletase oleks mugavamaks vaatamiseks eespool, siis aitab käsuahela lõpus select(keeletase, everything())

```
koos %>% select(-taht) %>% t() %>% as_tibble() %>% add_column(asukoht=gsub(".*/(.*).txt",
"\\1", asukohad), .before=1) %>% left_join(dokmeta %>% select(kood, keeletase),
by=c("asukoht"="kood")) %>% select(keeletase, everything())

# A tibble: 395 × 119
   keeletase               asukoht    V1    V2    V3    V4    V5    V6    V7    V8
       <chr>                 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1       <NA>             hsanomat1     6     1   134    10     9    13     3     4
2       <NA>             hsanomat2     0     1   176    12    17    14     0     4
3       <NA>            postimees1     3     1   405    22    33    35     0     0
4       <NA>            postimees2     5     0   380    23    36    26     2     0
5         B1 doc_104580264060_item     2     0    91     0     4     9     0     0
6         C1 doc_104580264061_item     9     0  1694     0   103   182     1     0
```

Edasi arvutame kõikidele tulpadele keeletaseme kaupa keskmised. Praegu loetellu jäänud ajaleheartiklid lähevad kokku teadmata keeletaseme alla ja eemaldame pärast.

```
koos %>% select(-taht) %>% t() %>% as_tibble() %>% add_column(asukoht=gsub(".*/(.*).txt",
"\\1", asukohad), .before=1) %>% left_join(dokmeta %>% select(kood, keeletase),
by=c("asukoht"="kood")) %>% select(keeletase, everything()) %>% group_by(keeletase) %>%
summarise_if(is.numeric, mean)

# A tibble: 10 × 118
   keeletase         V1         V2        V3       V4         V5        V6         V7
       <chr>      <dbl>      <dbl>     <dbl>    <dbl>      <dbl>     <dbl>      <dbl>
1          A  1.6750000 0.00000000  150.6500 0.000000   7.075000  21.50000  0.3750000
2         A1  0.0000000 0.00000000   61.0000 0.000000   6.000000  20.00000  0.0000000
3         A2  0.9019608 0.00000000  107.6471 0.000000   5.960784  16.19608  0.1764706
4          B  2.8666667 0.00000000  288.5556 0.000000  18.400000  23.73333  0.2888889
5         B1  1.2017544 0.00000000  183.2193 0.000000  12.333333  20.53509  0.7543860
6         B2  2.6266667 0.00000000  300.7200 0.000000  20.880000  27.04000  1.2000000
7          C  4.5000000 0.00000000  426.2500 0.000000  28.000000  37.00000  1.5000000
8         C1  7.7000000 0.00000000  744.9500 0.000000  53.900000  62.35000  2.8500000
9         C2 12.0000000 0.00000000 1479.0000 0.000000 182.000000 133.00000 26.0000000
10      <NA>  1.7500000 0.06818182  222.8409 1.522727  17.409091  28.06818  1.5000000
# ... with 110 more variables: V8 <dbl>, V9 <dbl>, V10 <dbl>, V11 <dbl>, V12 <dbl>,
```

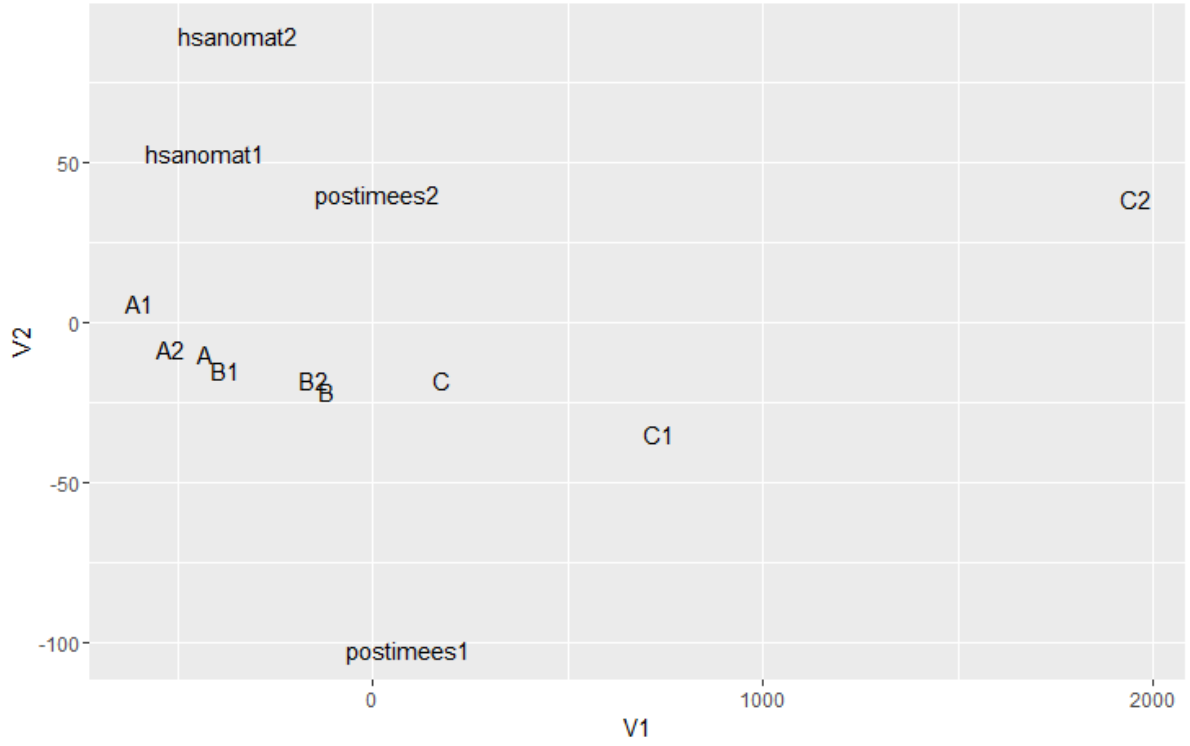
Kuna soovime aga ajalehtede andmeid võrdlusena näha, siis tuleb nad uuesti tabelisse lisada. Nende nähtava tulba nimeks on `asukoht`, keeletasemete keskmistel aga

`keeletase`. Nimetame õppijatekstide keeletaseme tulba ka asukohaks, nii saab tabelid teineteisele otsa liita - bind_rows käsu abil esialgses puhris olnud tabeli neli esimest rida.

```
> koos %>% select(-taht) %>% t() %>% as_tibble() %>% add_column(asukoht=gsub(".*/(.*).txt",
"\\1", asukohad), .before=1) %>% left_join(dokmeta %>% select(kood, keeletase),
by=c("asukoht"="kood")) %>% select(keeletase, everything()) %>% {. ->> puhver} %>%
group_by(keeletase) %>% summarise_if(is.numeric, mean) %>% ungroup() %>%
rename(asukoht=keeletase) %>% bind_rows(puhver %>% head(4))
# A tibble: 14 × 119
      asukoht         V1         V2        V3        V4         V5        V6
        <chr>      <dbl>      <dbl>     <dbl>     <dbl>      <dbl>     <dbl>
1           A  1.6750000 0.00000000  150.6500  0.000000   7.075000  21.50000
2          A1  0.0000000 0.00000000   61.0000  0.000000   6.000000  20.00000
3          A2  0.9019608 0.00000000  107.6471  0.000000   5.960784  16.19608
4           B  2.8666667 0.00000000  288.5556  0.000000  18.400000  23.73333
5          B1  1.2017544 0.00000000  183.2193  0.000000  12.333333  20.53509
6          B2  2.6266667 0.00000000  300.7200  0.000000  20.880000  27.04000
7           C  4.5000000 0.00000000  426.2500  0.000000  28.000000  37.00000
8          C1  7.7000000 0.00000000  744.9500  0.000000  53.900000  62.35000
9          C2 12.0000000 0.00000000 1479.0000  0.000000 182.000000 133.00000
10       <NA>  1.7500000 0.06818182  222.8409  1.522727  17.409091  28.06818
11  hsanomat1  6.0000000 1.00000000  134.0000 10.000000   9.000000  13.00000
12  hsanomat2  0.0000000 1.00000000  176.0000 12.000000  17.000000  14.00000
13 postimees1  3.0000000 1.00000000  405.0000 22.000000  33.000000  35.00000
14 postimees2  5.0000000 0.00000000  380.0000 23.000000  36.000000  26.00000
# ... with 112 more variables: V7 <dbl>, V8 <dbl>, V9 <dbl>, V10 <dbl>, V11 <dbl>,
```

Mitmemõõtmelise skaleerimise käsu `cmdscale` käivitamise tarbeks tuleb kõik mittearvulised tulbad eemaldada - praegu ajalehetekstide juures jäänud keeletaseme tulp. Asukohtade edasise kuvamise tarbeks salvestame selle muutujasse `puhver2` ja eemaldame asukoha käsuahelast. Arvutame kaugused objektide vahel ning kuvame tulemused endisel moel ekraanile

```
> koos %>% select(-taht) %>% t() %>% as_tibble() %>% add_column(asukoht=gsub(".*/(.*).txt",
"\\1", asukohad), .before=1) %>% left_join(dokmeta %>% select(kood, keeletase),
by=c("asukoht"="kood")) %>% select(keeletase, everything()) %>% {. ->> puhver} %>%
group_by(keeletase) %>% summarise_if(is.numeric, mean, na.rm=TRUE) %>% ungroup() %>%
rename(asukoht=keeletase) %>% bind_rows(puhver %>% head(4)) %>% select(-keeletase) %>%
na.omit() %>% {. ->> puhver2} %>% select(-asukoht) %>% dist() %>% cmdscale(2) %>%
as_tibble() %>% add_column(asukoht=puhver2$asukoht) %>% ggplot(aes(V1, V2, label=asukoht))
+ geom_text()
```

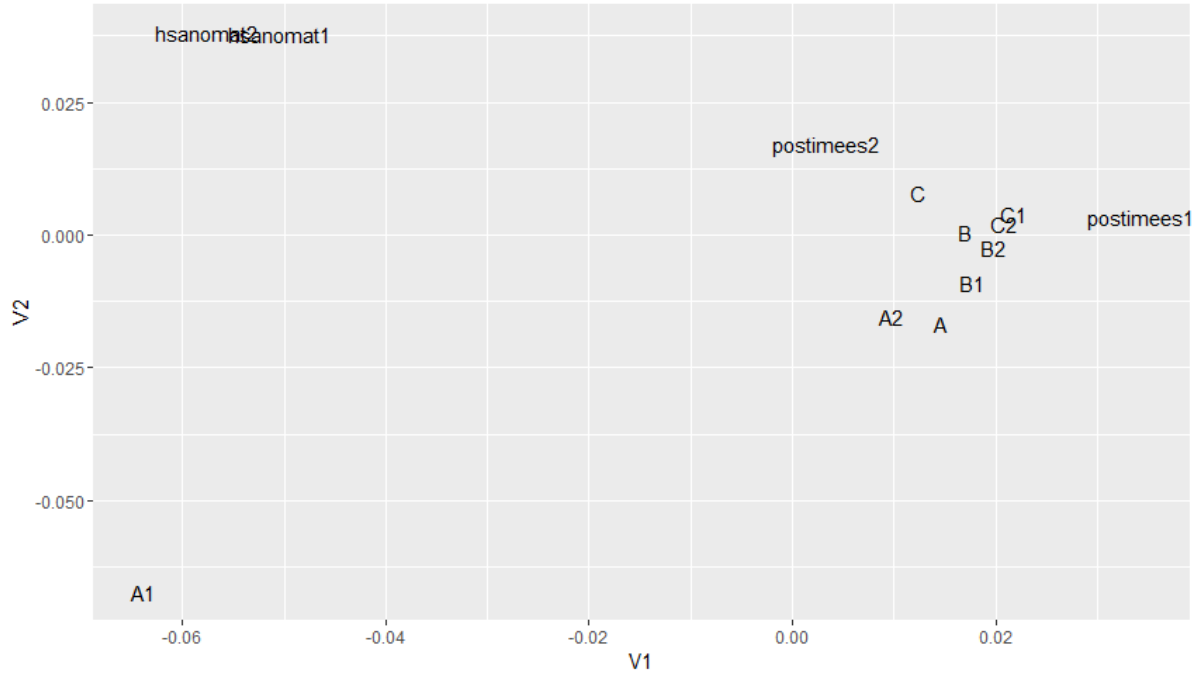


Nagu näha, siis paigutuvad õppijakeelte kõigi rühmade keskmised kahe Postimehe artikli vahele.

Tekstide keskmised paiknevad suhteliselt riburada pidi keeletasemete järgi. Vaadates tekib aga kahtlus, et ehk mõjutab seda paiknemist pigem teksti pikkus kui tähtede osakaal. Pikkuse eemaldamiseks jagame kõik read tabelis vastavate ridade tähtede arvu summadega ehk teksti pikkustega läbi.

```
koos %>% select(-taht) %>% t() %>% as_tibble() %>% {./rowSums(.)} %>%
add_column(asukoht=gsub(".*/(.*).txt", "\\1", asukohad), .before=1) %>% left_join(dokmeta
%>% select(kood, keeletase), by=c("asukoht"="kood")) %>% select(keeletase, everything())
%>% {. ->> puhver} %>% group_by(keeletase) %>% summarise_if(is.numeric, mean, na.rm=TRUE)
%>% ungroup() %>% rename(asukoht=keeletase) %>% bind_rows(puhver %>% head(4)) %>%
select(-keeletase) %>% na.omit() %>% {. ->> puhver2} %>% select(-asukoht) %>% dist() %>%
cmdscale(2) %>% as_tibble() %>% add_column(asukoht=puhver2$asukoht) %>% ggplot(aes(V1, V2,
label=asukoht)) + geom_text()
```

Pilt mõnevõrra muutus, aga õppijatekstide keskmised jäävad siiski kahe Postimehe artikli vahele



Harjutus

- Tehke näide läbi
- Koostage joonis, kus eri värvidega on märgitud ajalehetekstide asukohad, keeletasemete keskmised ning iga konkreetne tekst oma keeletaseme tähisega

```
tahtedeSagedused <- function(failinimi){
  tekst= read_file(failinimi)
  vastus=tibble(taht=str_split(str_to_lower(tekst), "")[[1]]) %>% group_by(taht) %>%
    summarise(kogus=n())
  return (vastus)
}


kataloog="http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/"
failinimed=c("hsanomat1.txt", "hsanomat2.txt", "postimees1.txt", "postimees2.txt")
asukohad=paste(kataloog, failinimed, sep="")

dokmeta=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokmeta.txt")

kataloog2="http://www.tlu.ee/~jaagup/andmed/keel/korpus/dok/"
failinimed2=dokmeta %>% filter(emakeel=="soome") %>% .$kood
asukohad2=paste(kataloog2, failinimed2, ".txt", sep="")

asukohad = c(asukohad, asukohad2)

koos=tahtedeSagedused(asukohad[1])
colnames(koos)=c("taht", basename(asukohad[1]))
for(failinimi in asukohad[2:length(asukohad)]){
  tabel=tahtedeSagedused(failinimi)
  colnames(tabel)=c("taht", basename(failinimi))
  koos=koos %>% full_join(tabel, by="taht")
}
koos=koos %>% replace(., is.na(.), 0)
```

```
head(koos)

lehed_tekstid=koos %>% select(-taht) %>% t() %>% as_tibble() %>%
  add_column(asukoht=gsub(".*/(.*).txt", "\\1", asukohad)) %>%
  add_column(tyyp=c(rep("ajaleht", length(failinimed)),
                    rep("oppijatekst", nrow(.)-length(failinimed)))) %>%
  left_join(dokmeta %>% select(kood, keeletase), by=c("asukoht"="kood")) %>%
  select(tyyp, keeletase, asukoht, everything())
```

```
> lehed_tekstid
# A tibble: 395 x 120
   tyyp  keeletase asukoht    V1    V2    V3    V4    V5    V6    V7    V8    V9
   <chr> <chr>     <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 ajal~ NA        hsanom~     6     1   134    10     9    13     3     4     3
 2 ajal~ NA        hsanom~     0     1   176    12    17    14     0     4     0
 3 ajal~ NA        postim~     3     1   405    22    33    35     0     0     3
 4 ajal~ NA        postim~     5     0   380    23    36    26     2     0     6
 5 oppi~ B1        doc_10~     2     0    91     0     4     9     0     0     4
 6 oppi~ C1        doc_10~     9     0  1694     0   103   182     1     0    29
 7 oppi~ B2        doc_10~     2     0   342     0    33    28     0     0     1
 8 oppi~ NA        doc_10~     0     0   213     0    14    22     0     0     0
 9 oppi~ C1        doc_10~    13     0   779     0    77    54     3     0     1
10 oppi~ B1        doc_10~     0     0    93     0     7    13     1     0     0
# ... with 385 more rows, and 108 more variables: V10 <dbl>, V11 <dbl>, V12 <dbl>,
```

Eraldi arvutus tähtede keskmise koguse kohta keeletasemete kaupa

```
tasekesk=lehed_tekstid %>% filter(tyyp=="oppijatekst")  %>% group_by(keeletase) %>%
summarise_if(is_numeric, mean, na.rm=TRUE)
tasekesk
```

```
> tasekesk
# A tibble: 10 x 118
   keeletase     V1    V2    V3    V4     V5    V6     V7    V8      V9    V10    V11
   <chr>      <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>  <dbl>  <dbl>
 1 A          1.68      0  151.     0   7.08  21.5  0.375     0  1.9     1.18   0.4
 2 A1         0         0   61      0   6     20    0         0  0       0      0
 3 A2         0.902     0  108.     0   5.96  16.2  0.176     0  0.0588  0.157  0.118
 4 B          2.87      0  289.     0  18.4   23.7  0.289     0  4.09    2.16   0.756
 5 B1         1.20      0  183.     0  12.3   20.5  0.754     0  0.351   0.211  0.193
 6 B2         2.63      0  301.     0  20.9   27.0  1.2       0  0.96    0.667  0.32
 7 C          4.5       0  426.     0  28     37    1.5       0  7.75    3.75   1.5
 8 C1         7.7       0  745.     0  53.9   62.4  2.85      0  7.35    2.8    1.15
 9 C2        12         0 1479      0 182    133   26         0 22      18     12
10 NA         1.58      0  218.     0  16.8   28.7  1.52      0  3.5     2.48   1.5
# ... with 106 more variables: V12 <dbl>, V13 <dbl>, V14 <dbl>, V15 <dbl>,
#   V16 <dbl>, V17 <dbl>, V18 <dbl>, V19 <dbl>, V20 <dbl>, V21 <dbl>, V22 <dbl>,
```

Lisame eelnevale ajalehtede ja õppijatekstide lehele read keeletasemete keskmiste tähtede arvudega. Keskmiste tabelile lisame tulba tyyp (nagu teiselgi tabelil), kuhu iga rea kohale kirjutatakse väärtus "tasek".

Alguses ajalehetekstid, järgnevad õppijakeeletekstid

```
lehed_tekstid_tasemed=lehed_tekstid %>%
   bind_rows(tasekesk %>% add_column(tyyp=rep("tasek", nrow(.))))
> head(lehed_tekstid_tasemed)
# A tibble: 6 x 120
  tyyp  keeletase asukoht    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10
  <chr> <chr>     <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 ajal~ NA        hsanom~     6     1   134    10     9    13     3     4     3     3
2 ajal~ NA        hsanom~     0     1   176    12    17    14     0     4     0     0
3 ajal~ NA        postim~     3     1   405    22    33    35     0     0     3     2
4 ajal~ NA        postim~     5     0   380    23    36    26     2     0     6     1
5 oppi~ B1        doc_10~     2     0    91     0     4     9     0     0     4     2
6 oppi~ C1        doc_10~     9     0  1694     0   103   182     1     0    29    12
# ... with 107 more variables: V11 <dbl>, V12 <dbl>, V13 <dbl>, V14 <dbl>,
#   V15 <dbl>, V16 <dbl>, V17 <dbl>, V18 <dbl>, V19 <dbl>, V20 <dbl>, V21 <dbl>,
```

ning pika tabeli lõppu tulevad keeletasemete keskmised väärtused vastava tähe arvu kohta tekstis

```
> tail(lehed_tekstid_tasemed, 15)
# A tibble: 15 x 120
   tyyp  keeletase asukoht     V1    V2    V3    V4     V5    V6     V7    V8       V9
   <chr> <chr>     <chr>    <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>
 1 oppi~ B         doc_91~  2          0  220      0   9     25    0          0  1
 2 oppi~ B         doc_91~  0          0  177      0   6     13    0          0  0
 3 oppi~ A         doc_98~  2          0  148      0   2     22    0          0  5
 4 oppi~ A         doc_98~  1          0   54      0   2      8    0          0  3
 5 oppi~ A         doc_98~  0          0  127      0   9     13    9          0  0
 6 tasek A         NA       1.68       0  151.     0   7.08  21.5  0.375      0  1.9
 7 tasek A1        NA       0          0   61      0   6     20    0          0  0
 8 tasek A2        NA       0.902      0  108.     0   5.96  16.2  0.176      0  0.0588
 9 tasek B         NA       2.87       0  289.     0  18.4   23.7  0.289      0  4.09
10 tasek B1        NA       1.20       0  183.     0  12.3   20.5  0.754      0  0.351
11 tasek B2        NA       2.63       0  301.     0  20.9   27.0  1.2        0  0.96
12 tasek C         NA       4.5        0  426.     0  28     37    1.5        0  7.75
13 tasek C1        NA       7.7        0  745.     0  53.9   62.4  2.85       0  7.35
14 tasek C2        NA      12          0 1479      0 182    133   26          0 22
15 tasek NA        NA       1.58       0  218.     0  16.8   28.7  1.52       0  3.5
```

Edasi arvutame multidimensionaalse skaleerimise kaudu koordinaadid.

```
koordinaadid=lehed_tekstid_tasemed %>% select_if(is_numeric) %>% dist() %>% cmdscale(2) %>%
as_tibble()

> head(koordinaadid)
# A tibble: 6 x 2
       V1       V2
    <dbl>    <dbl>
1  -180.    11.5
2   -98.7    7.15
3   333.   -43.0
4   255.   -29.7
5  -317.     0.667
6  3006.   398.
```

Koordinaatidele näitamise tarbeks juurde rea tüüp, ja silt. Esialgu sildiks vastava rea keeletase, ajalehtede puhul artikli nimetus.

Lehe numbrite arvutamisel

```
lehenrd=which(lehed_tekstid_tasemed$tyyp=="ajaleht")
```

annab välja, milliste järjekorranumbritega kirjed on ajalehtede omad

```
> lehenrd
[1] 1 2 3 4
```

Lõpus ka puuduvate keeletasemetega read välja

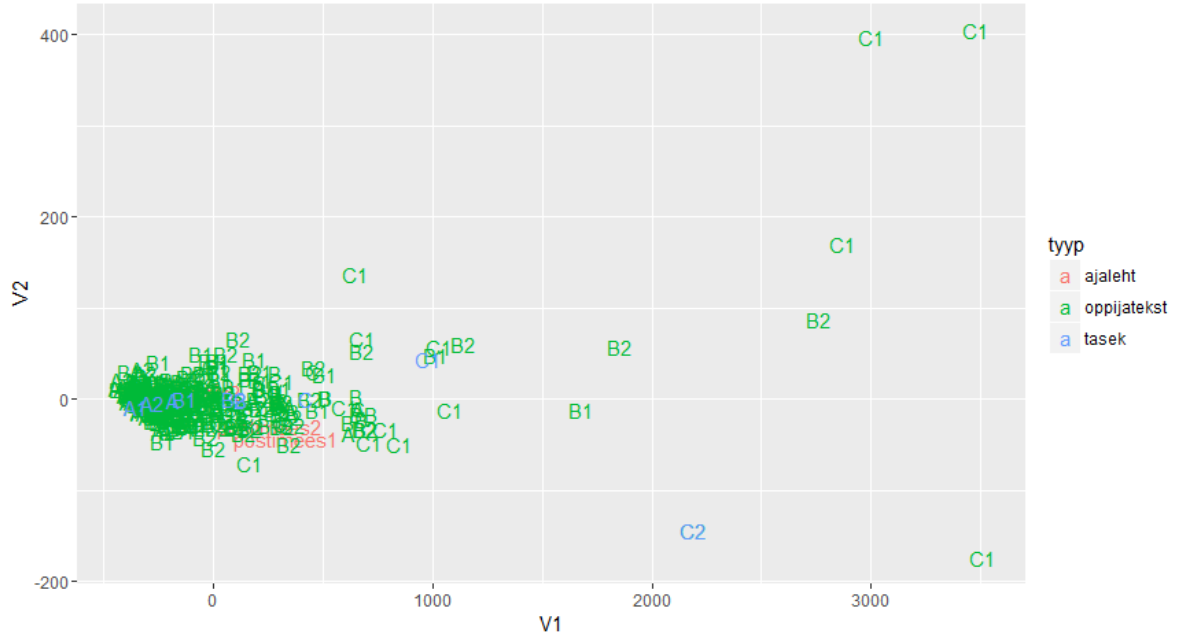
```
koordinaadid$tyyp=lehed_tekstid_tasemed$tyyp
koordinaadid$silt=lehed_tekstid_tasemed$keeletase
lehenrd=which(lehed_tekstid_tasemed$tyyp=="ajaleht")
koordinaadid[lehenrd, "silt"]=lehed_tekstid_tasemed[lehenrd, "asukoht"]
koordinaadid=na.omit(koordinaadid)
```

Teisenduste tulemus:

```
> koordinaadid
# A tibble: 364 × 4
           V1          V2        tyyp       silt
        <dbl>       <dbl>       <chr>      <chr>
1  -179.62711  11.5420155     ajaleht  hsanomat1
2   -98.68463   7.1538032     ajaleht  hsanomat2
3   333.31699 -42.9937681     ajaleht postimees1
4   254.56353 -29.7205308     ajaleht postimees2
5  -316.55116   0.6670346 oppijatekst         B1
6  3005.86663 397.9367684 oppijatekst         C1
7   180.02208  -9.4929439 oppijatekst         B2
8  1035.05123  56.7203021 oppijatekst         C1
9  -285.95881  19.1694807 oppijatekst         B1
10 -289.01532   8.7155037 oppijatekst         B1
# ... with 354 more rows
```

Tabeli järgi joonis

```
koordinaadid %>% ggplot(aes(V1, V2, label=silt, color=tyyp)) + geom_text()
```



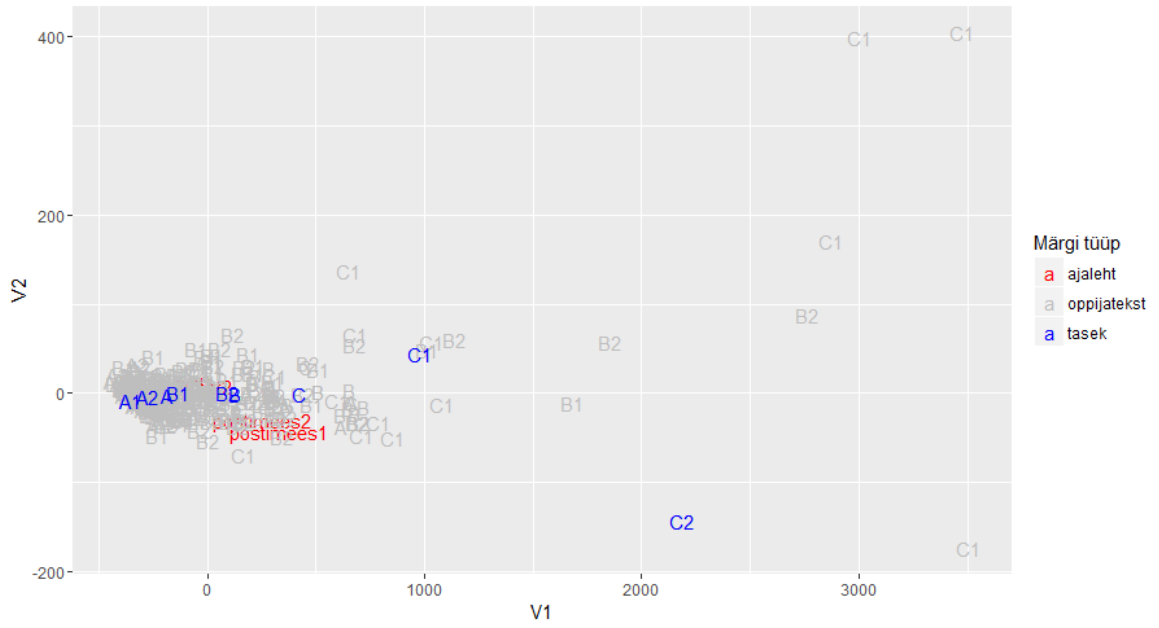
Joonise täiendusi

Enamvähem näeb, kus paiknevad ajalehed, kus tasemed ja kus tekstid, aga kuna tekste palju ja nad erksalt rohelised, siis muud jäävad nende varju. Värvide ümber arvutuseks loon muutuja, kus kirjas millise tooniga milline tüüp on

```
toonid=c("ajaleht"="red", "oppijatekst"="gray", "tasek"="blue")
```

Nii saab tekstide andmed õrnalt paistvaks halliks

```
koordinaadid %>% ggplot(aes(V1, V2, label=silt, color=tyyp)) + geom_text() +
scale_color_manual(name="Märgi tüüp", values=toonid)
```

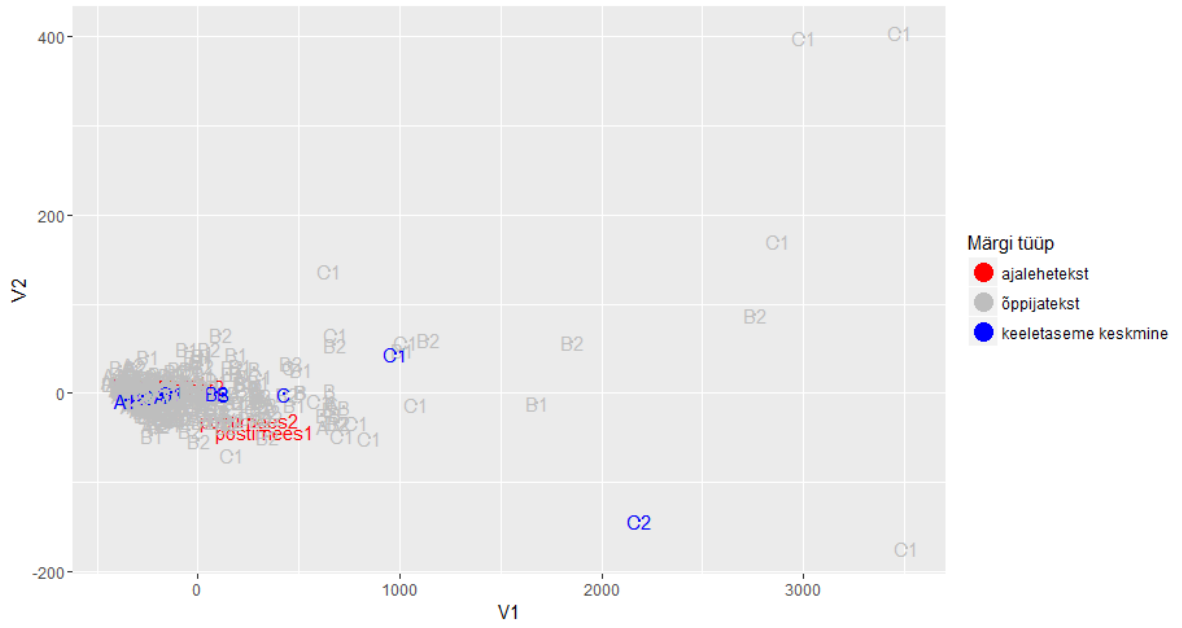


Käsu `scale_color_manual` juures `labels`-atribuudiga võib määrata legendisiltide tekstid.

Joonistatud tekstid näitavad legendil endid a-tähelise ikooniga. Selle asendamiseks silmale veidi rahulikuma ringiga on moodus, kus lisaks tekstidele joonistatakse ka nende asukohtade punktid ekraanile, aga suurusega 0 ning tekstide puhul määratakse, et legendi ei näidata.

Mummude suuruse pääseb paika sättima guides-alt pika käsu kaudu.

```
> koordinaadid %>% ggplot(aes(V1, V2, label=silt, color=tyyp)) + geom_text(show.legend=F) +
scale_color_manual(name="Märgi tüüp", values=toonid, labels=c("ajalehetekst",
"õppijatekst", "keeletaseme keskmine")) + geom_point(size=0) +  guides(colour =
guide_legend(override.aes = list(size=5)))
```

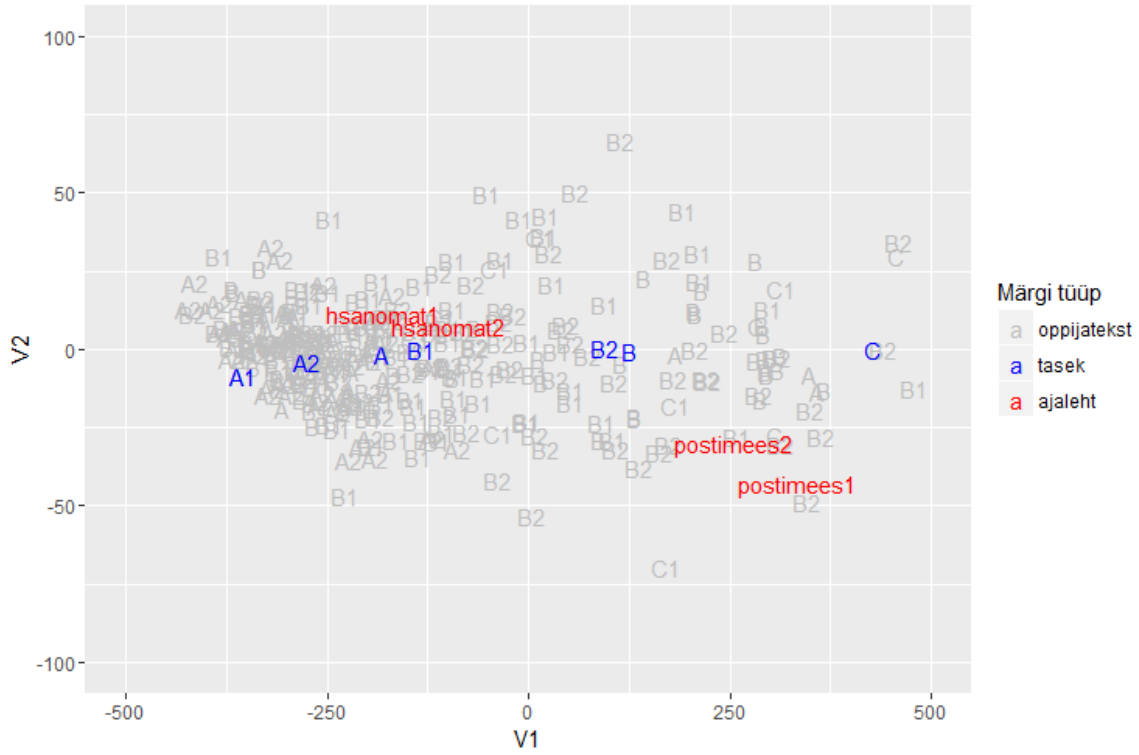


Praegu joonistades jäävad ajalehetekstid kõige alla, sest nad on koordinaatide loetelus esimesed. Parema vaatamis huvides püüame aga taustaks olevad üksikud tekstid kõige ette jätta. Selleks arvutame tüübi ümber faktoriks, kus tasemete järjekorras on esimene õppijatekst, siis tuleb keeletaseme keskmine ning alles lõpuks ajalehetekst. Järjestame tabeli selle järgi

```
> koordinaadid %>% mutate(tyyp=factor(tyyp, levels=c("oppijatekst", "tasek", "ajaleht")))
%>% arrange(tyyp)
# A tibble: 364 × 4
          V1          V2        tyyp  silt
       <dbl>       <dbl>      <fctr> <chr>
1  -316.5512   0.6670346 oppijatekst    B1
2  3005.8666 397.9367684 oppijatekst    C1
3   180.0221  -9.4929439 oppijatekst    B2
4  1035.0512  56.7203021 oppijatekst    C1
```

Nüüd joonise luues ning sobivate tunnuste järgi välja näidates jäävad õppijatekstid selgelt tahaplaanile ning ajalehed ja keeletasemete keskmised paistavad paremini välja.

```
> koordinaadid %>% mutate(tyyp=factor(tyyp, levels=c("oppijatekst", "tasek", "ajaleht")))
%>% arrange(tyyp) %>% ggplot(aes(V1, V2, label=silt, color=tyyp)) + geom_text() +
scale_color_manual(name="Märgi tüüp", values=toonid) + xlim(-500, 500) +ylim(-100, 100)
```



Harjutus

- Ühenda eelmiste seletuste võimalused - järjesta õppijatekstid alla, kuva tüüpide nimed legendis pikema seletava tekstiga ning asenda a-tähed mummudega

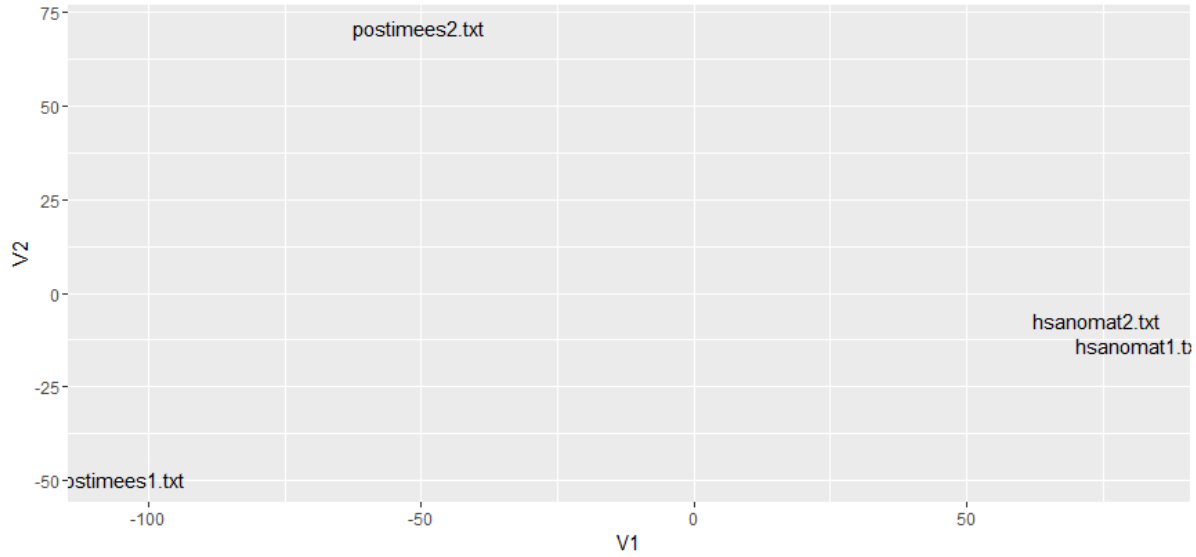
Tähepaarid ja MDS

Üksikutest tähtedest enam aitavad mustreid välja tuua tähepaarid, samuti kolmikud ja nelikud. Samas pikemate jadade võrdlemiseks peab olema enam materjali, sest võimalikke väärtusi tekib palju rohkem ning sealtkaudu jääb igale järjendile vähem leidumiskohti. Paljud järjendid jäävad nõnda hoopis ainukordseteks ning nende kaudu ei saa võrreldavate dokumentide eripärasid ja sarnasusi välja tuua. Tähepaarid on nõnda pärast ühetähelisi võrdlusi järgmine valik.

```
failinimi<- "http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/postimees1.txt"
tekst <- read_file(failinimi) %>% str_to_lower() %>% str_replace_all("[^a-zõäöü]", "")
sapply(1:(str_length(tekst)-1), function(koht){str_sub(tekst, koht, koht+1)})
```

```
koos=sagedused(asukohad[1])
colnames(koos)=c("vaartus", basename(asukohad[1]))
for(failinimi in asukohad[2:length(asukohad)]){
  tabel=sagedused(failinimi)
```

```
  colnames(tabel)=c("vaartus", basename(failinimi))
  koos=koos %>% full_join(tabel, by="vaartus")
}
koos=koos %>% replace(., is.na(.), 0)

koos %>% select(-vaartus) %>% t() %>% dist() %>% cmdscale(2) %>% as_tibble() %>%
add_column(failinimi=failinimed) %>% ggplot(aes(V1, V2, label=failinimi)) + geom_text()
```



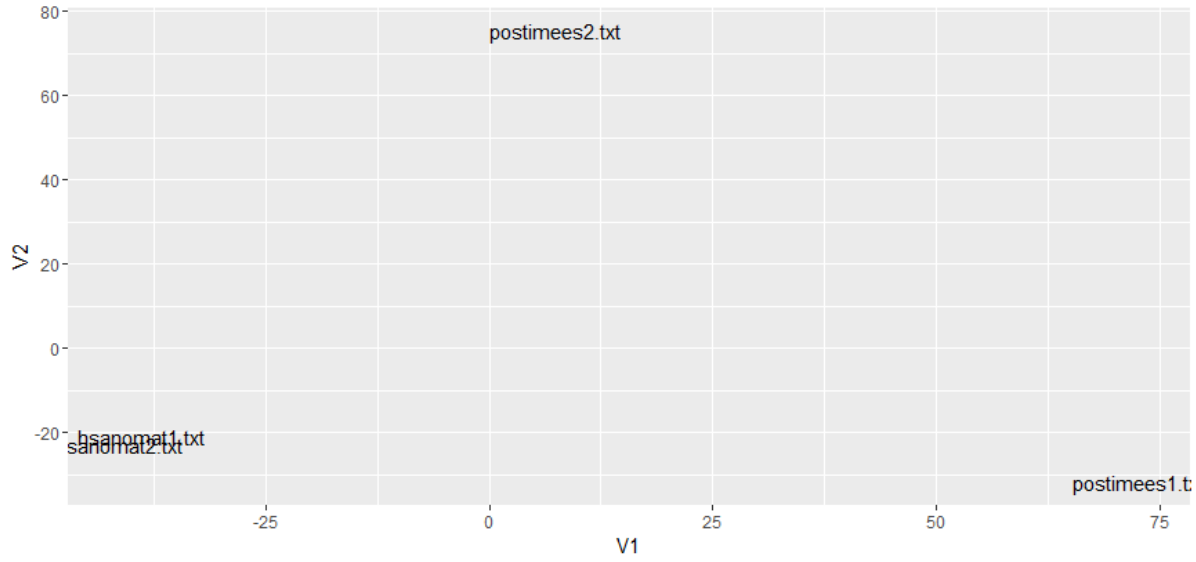
Loetelust paistab nõnda, et milliseid tähepaare kuivõrd esineb.

```
> koos
# A tibble: 418 × 5
   vaartus hsanomat1.txt hsanomat2.txt postimees1.txt postimees2.txt
     <chr>         <dbl>         <dbl>          <dbl>          <dbl>
1       aa            12            16             23             20
2       ab             2             1              8              4
3       ae             4             3              8             11
4       ah             3             4              8             12
5       ai            17            14              2             14
6       aj             5             2             19              7
7       ak             2             5             16             24
8       al             8            15             50             38
9       am             4            10             26             12
10      an            17            14             24             20
# ... with 408 more rows
```

Kolmikute puhul on arvud juba märgatavalt väiksemad ning kolmikuid endid ligi viis korda rohkem. Samas sealtkaudu paistab välja, et soomekeelsed tekstid on veel märgatavalt enam teineteisele sarnased

```
vaartus=sapply(1:(str_length(tekst)-2), function(koht){str_sub(tekst, koht, koht+2)})

> koos
# A tibble: 2,217 × 5
   vaartus hsanomat1.txt hsanomat2.txt postimees1.txt postimees2.txt
     <chr>         <dbl>         <dbl>          <dbl>          <dbl>
1      aai             1             0              0             10
2      aaj             2             1              0              0
```

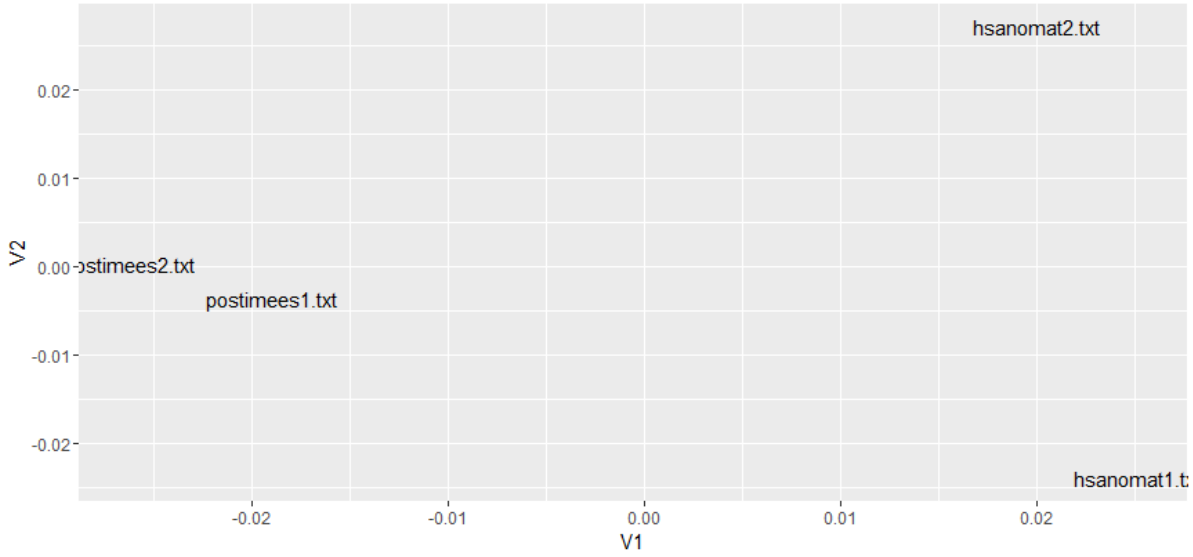
```
3       aan             3             3             3             3
4       aar             1             1             3             0
5       aas             2             4             8             0
6       aat             2             0             3             1
7       aav             1             2             1             0
8       abr             2             0             0             0
9       aeh             1             0             1             3
10      aen             1             2             0             0
# ... with 2,207 more rows
```



Kui jagada iga rea väärtus rea summaga ehk vastava kolmiku üldarvuga tekstides läbi, siis kahaneb algsete suurte väärtuste osa ning märgile pääsevad ka harvem esinevad kuid suurte vahedega tunnused.

```
koos %>% select(-vaartus) %>% t() %>% {./rowSums(.)} %>% dist() %>% cmdscale(2) %>%
as_tibble() %>% add_column(failinimi=failinimed) %>% ggplot(aes(V1, V2, label=failinimi)) +
geom_text()
```

Selle pildi järgi satuvad hoopis kaks eestikeelset teksti omavahel rohkem lähestikku. Uurija ülesandeks ongi meetodeid katsetada mitmel moel ning leida moodused, mis parajasti uuritavat omadust võimalikult selgelt välja näitavad.



Stilomeetria

Ehk vahendite komplekt, mille abil püütakse hinnata tekstide sarnasust ning seeläbi vahel aimata, et millal võiks tegemist olla sama või sarnase autoriga. Meetodeid saab kõiki ükshaaval ja põhjalikumalt kasutada, aga R-keeles on nende põhivõimalused koondatud paketti nimega stylo, mis kasulik eraldi installeerida ning sisse lugeda

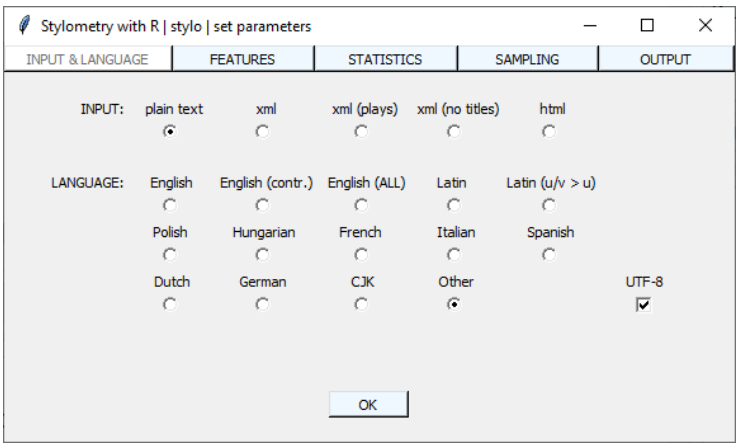
```
install.packages("stylo")
library(stylo)
```

Näiteandmetena juba tuttavad neli ajaleheteksti
`postimees1.txt postimees2.txt hsanomat1.txt hsanomat2.txt`
kataloogist
`http://www.tlu.ee/~jaagup/andmed/keel/ajalehetekstid/`
salvestatuna kataloogi
`d:\jaagup\lehetekstid\corpus`
Pakett `stylo` leiab tekstid alamkataloogist `corpus`, nii tuleb põhikaust määrata käsuga
`setwd("d:/jaagup/lehetekstid/")`

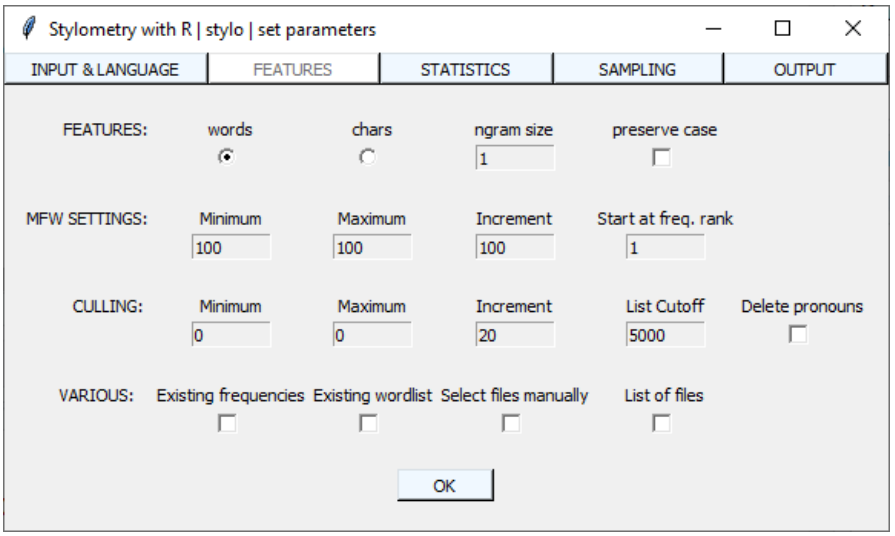
Erinevalt R-i tavapärasest tekstipõhisest lähenemisest avaneb käsu käivitamisel siin graafiline valikute aken

```
> stylo()
```

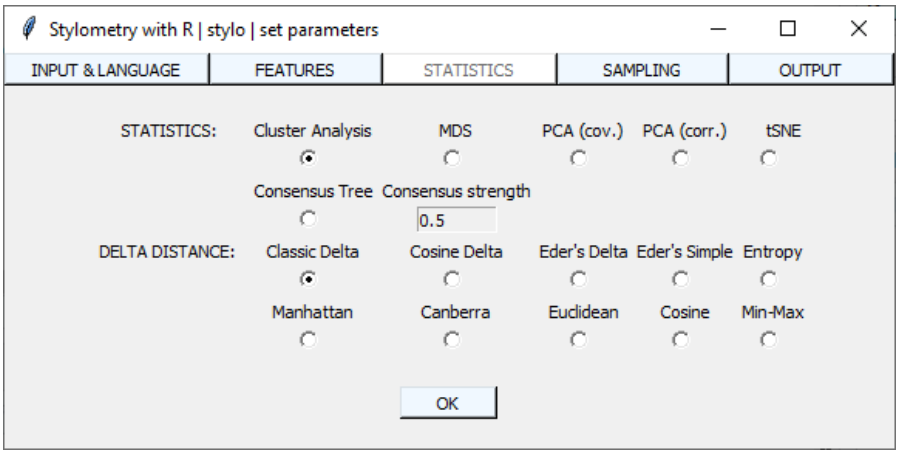
Kõigepealt sisendi tüübi valik. Lihtsaimal juhul paljas lihttekst.



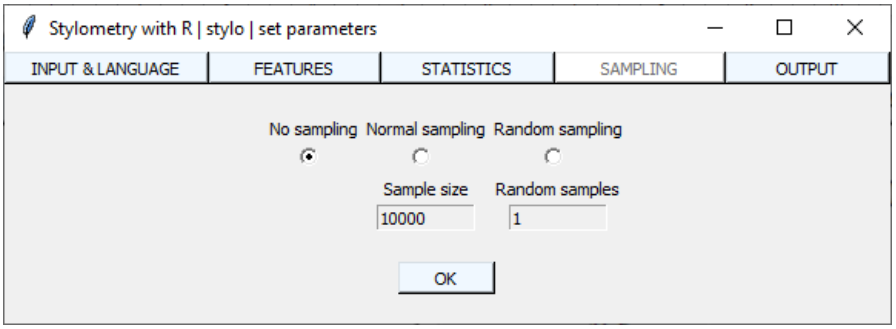
Järgmise saki peal valik, et mida uuritakse. Jätame praegu, et sõnu ja ühekaupa



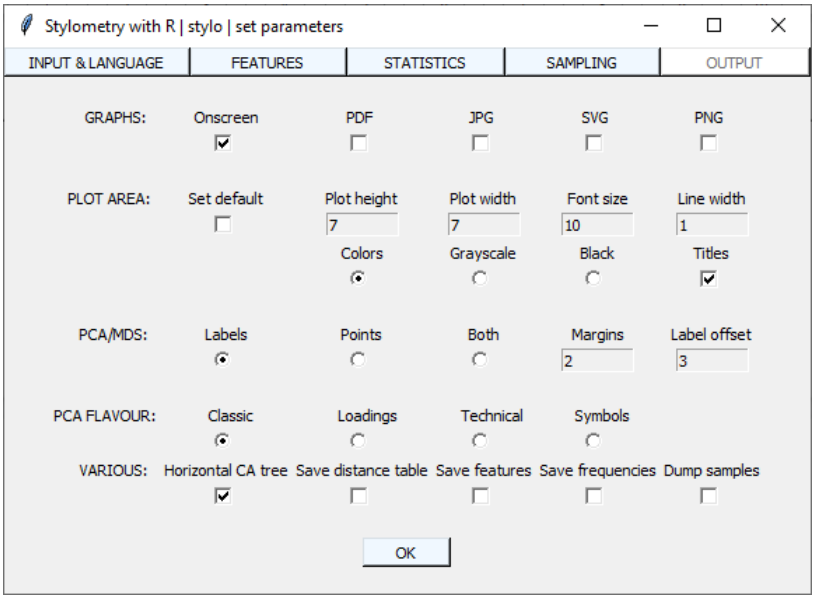
Meetodi valik - esimene võimalus on hierarhiline klasteranalüüs, ehk läheduste puu



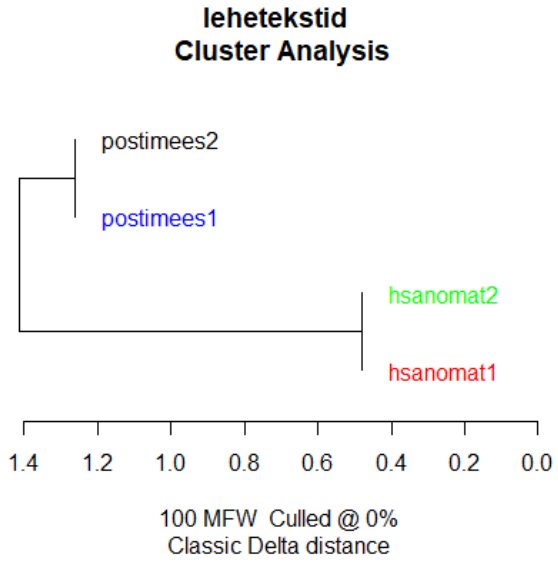
Kuna suhteliselt lühikesed tekstid, siis neid praegu omakorda tükeldada pole vaja



Ka viimase saki võib esialgu paika jätta. Ehk siis soovime väljundit ekraanile ning eraldi faili sisse ei telli.



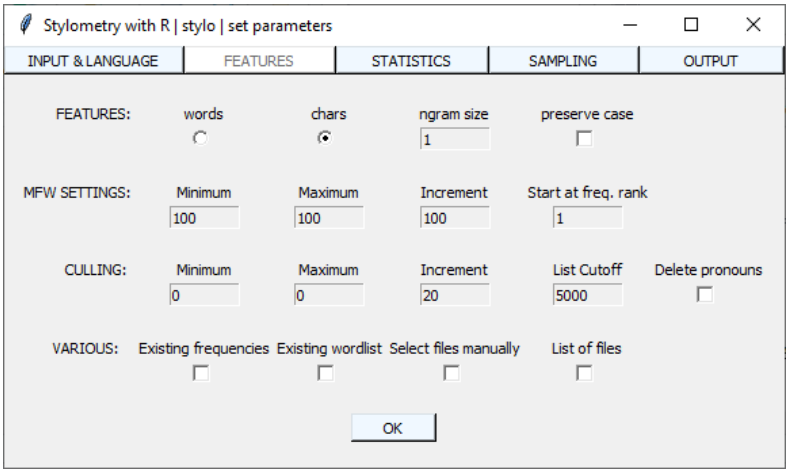
Tulemuseks joonis, kus kaks Postimehe teksti on ühes ja Helsingin Sanomate teksti teises harus. Kusjuures sõnade kaupa uurides satuvad soomekeelsed tekstid teineteisele märgatavalt lähemale



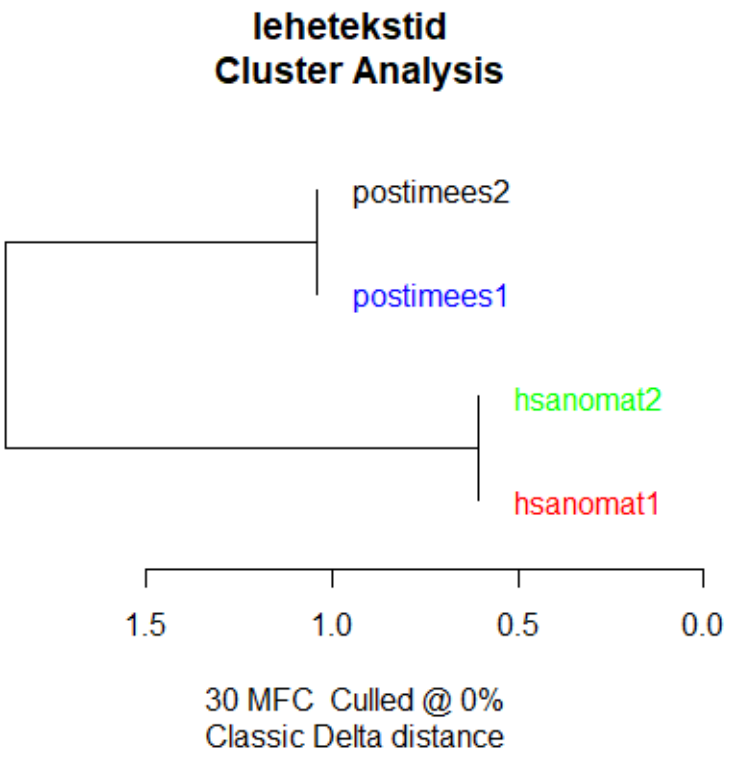
Käsu uus käivitus

```
> stylo()
```

ning katsel uurime tähtede sagedusi

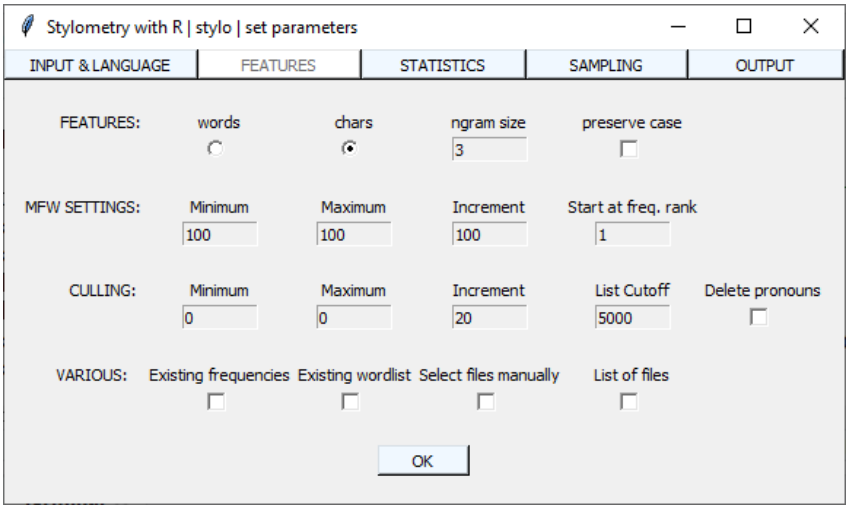


Ikka keele kaupa paaris, aga soomekeelsete tekstide omavaheline lähedus ei ületa enam nii tugevalt eestikeelsete oma

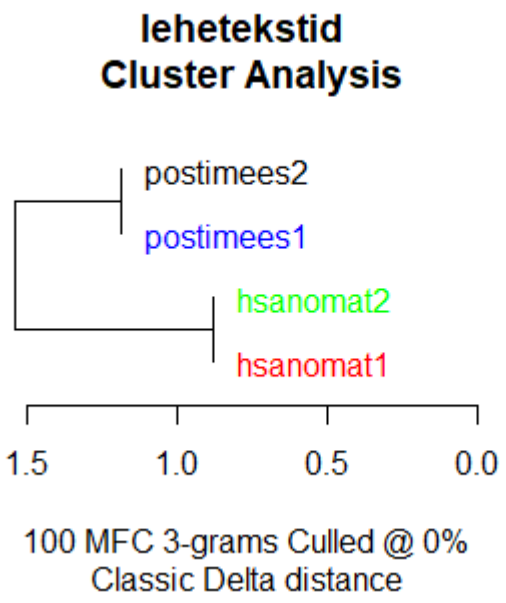


Järgmisel käivitusel võrdlus kolmetäheliste jadade kaupa

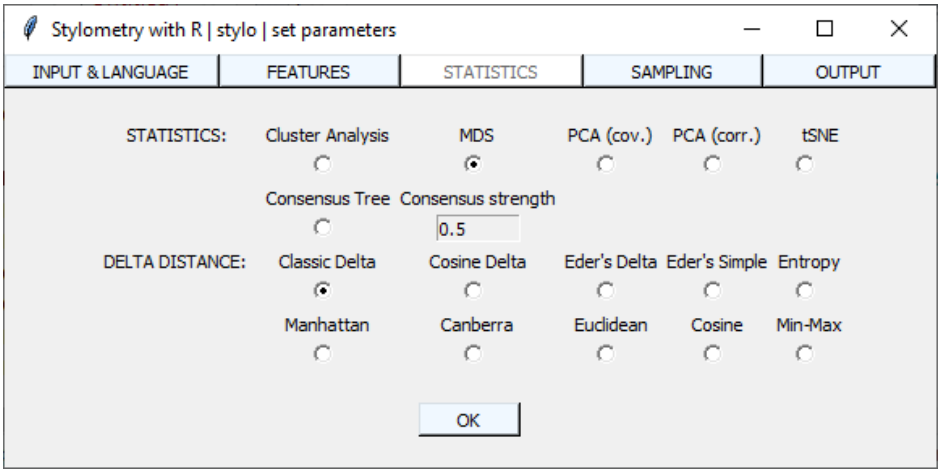
```
stylo()
```



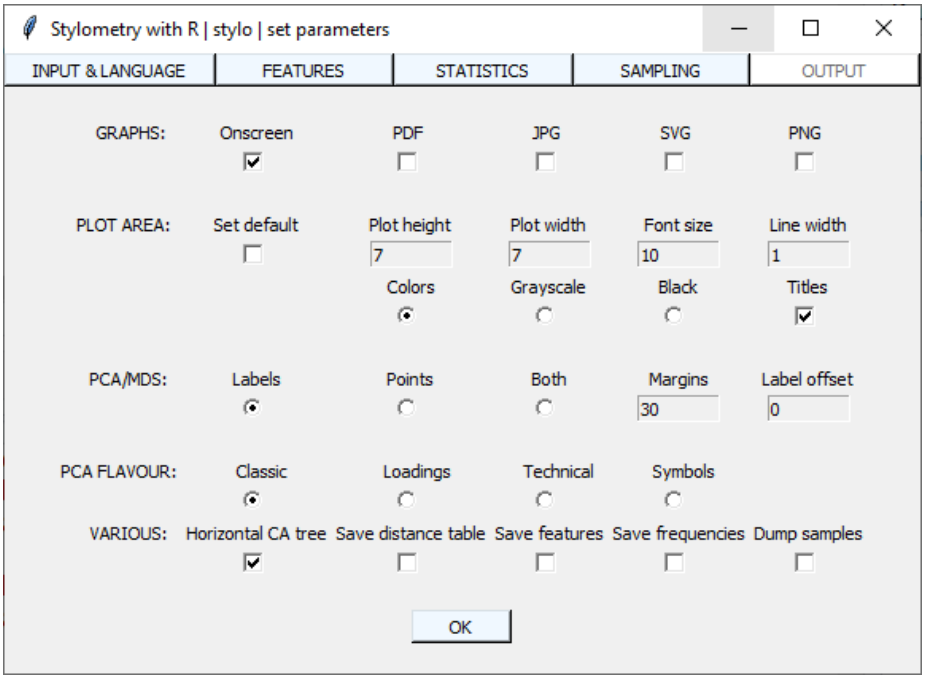
Ei tule ka nii suurt vahet sisse



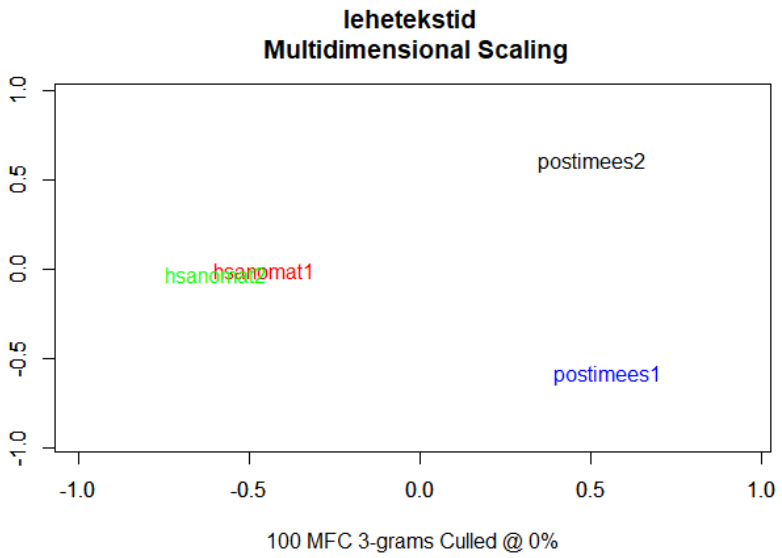
Nüüd meetodiks multidimensionaalne skaleerimine



Ning joonisel servadesse veidi rohkem ruumi, et kõik sõnad ära mahuksid

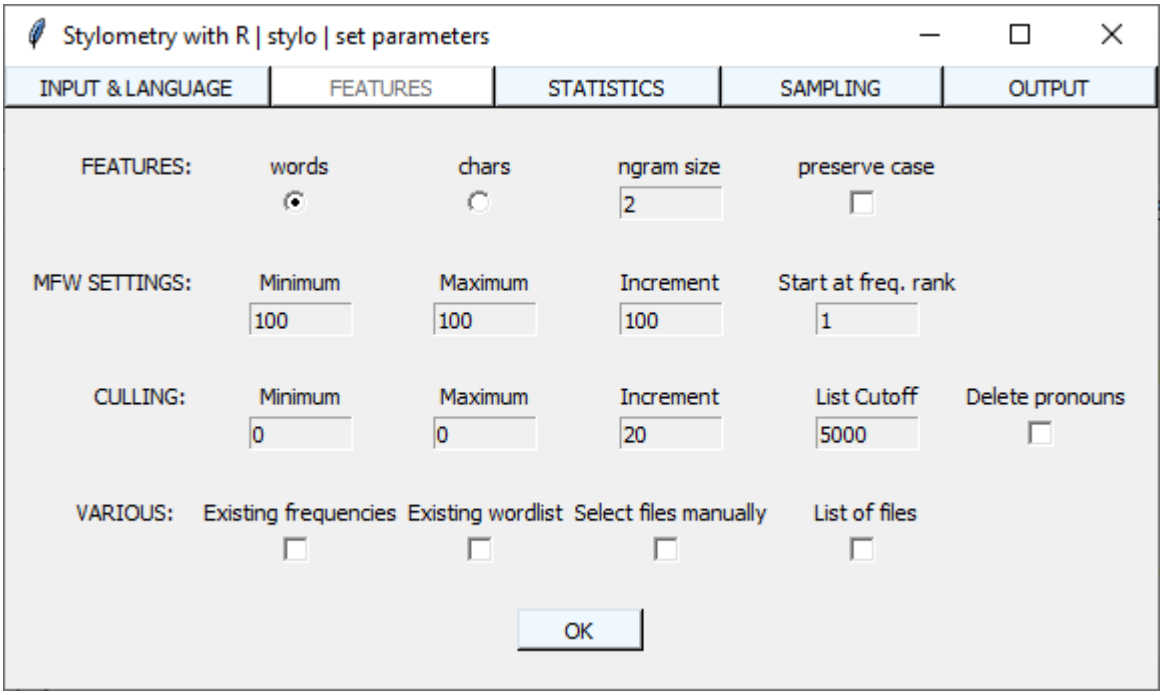


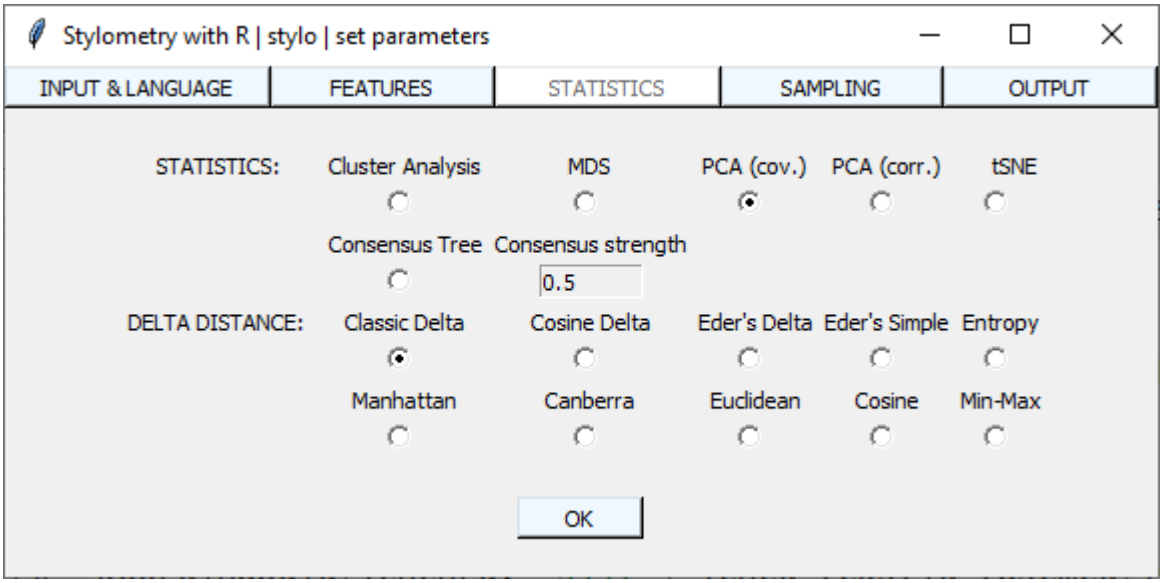
Tulemuseks tekstide asetumine kahemõõtmelisel skaalal, nii nagu neid varemgi sama meetodi juures näha võiks.

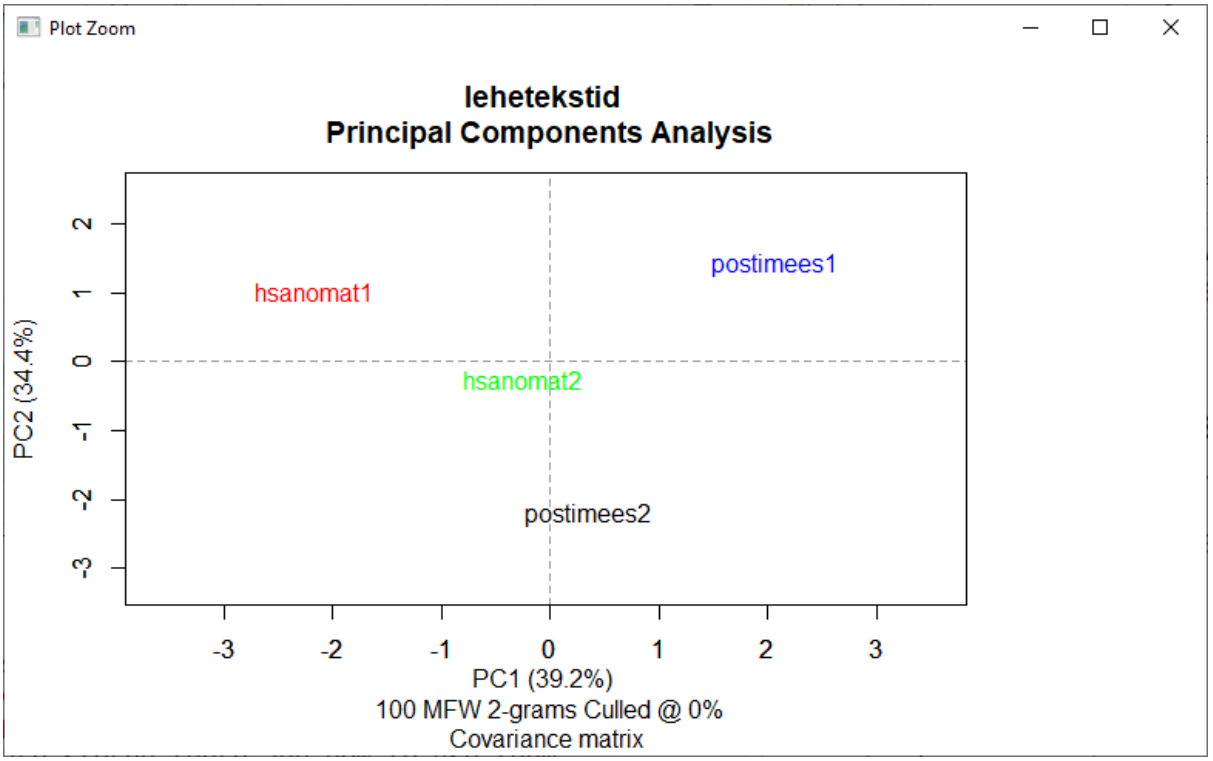


Harjutus

- Leidke samade tekstide paigutus joonisel kasutades peakomponentide analüüsi, arvestades kahesõnalisi järgnevusi

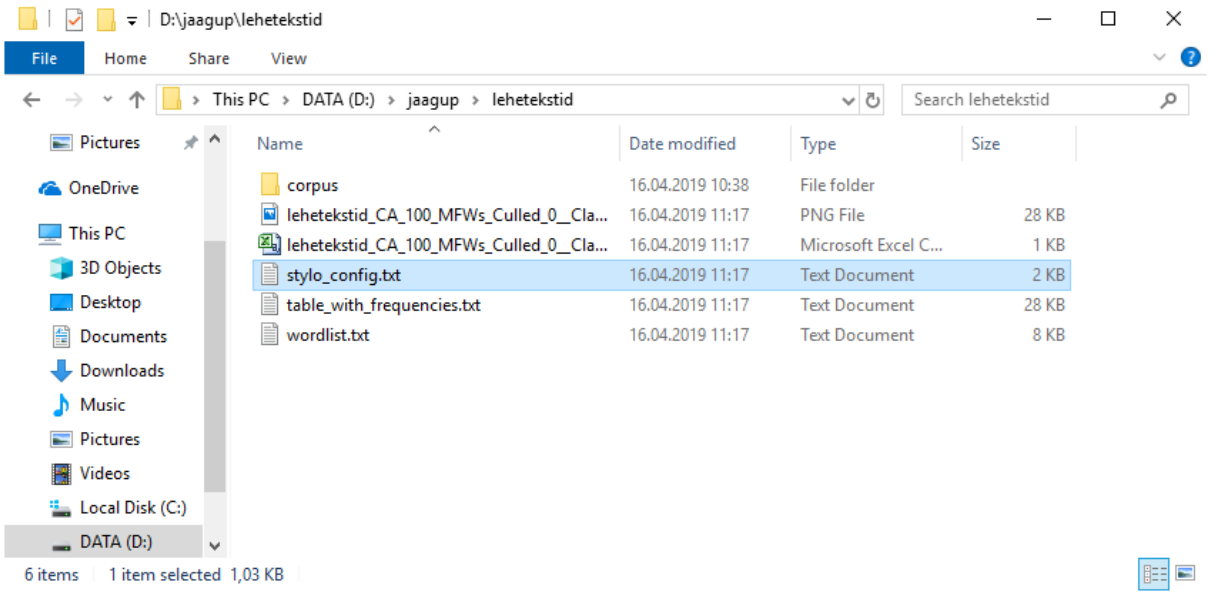

Stylometry with R | stylo | set parameters
INPUT & LANGUAGE
FEATURES
STATISTICS
SAMPLING
OUTPUT
FEATURES:
words
chars
ngram size
2
preserve case
MFW SETTINGS:
Minimum
100
Maximum
100
Increment
100
Start at freq. rank
1
CULLING:
Minimum
0
Maximum
0
Increment
20
List Cutoff
5000
Delete pronouns
VARIOUS:
Existing frequencies
Existing wordlist
Select files manually
List of files
OK


Stylometry with R | stylo | set parameters
INPUT & LANGUAGE
FEATURES
STATISTICS
SAMPLING
OUTPUT
STATISTICS:
Cluster Analysis
MDS
PCA (cov.)
PCA (corr.)
tSNE
Consensus Tree
Consensus strength
0.5
DELTA DISTANCE:
Classic Delta
Cosine Delta
Eder's Delta
Eder's Simple
Entropy
Manhattan
Canberra
Euclidean
Cosine
Min-Max
OK



Vaikesätete järgi vastuse küsides võib määrata parameetri gui=FALSE, sellisel juhul ei avata eraldi akent valikuteks. Kui tahta näiteks PNG-vormingus vastusefaili saada, tuleb see parameetrina käsklusesse lisada. Mis parameetreid veel pruukida saab, näeb failist stylo_config.txt

```
> stylo(gui=FALSE, write.png.file = TRUE)
```



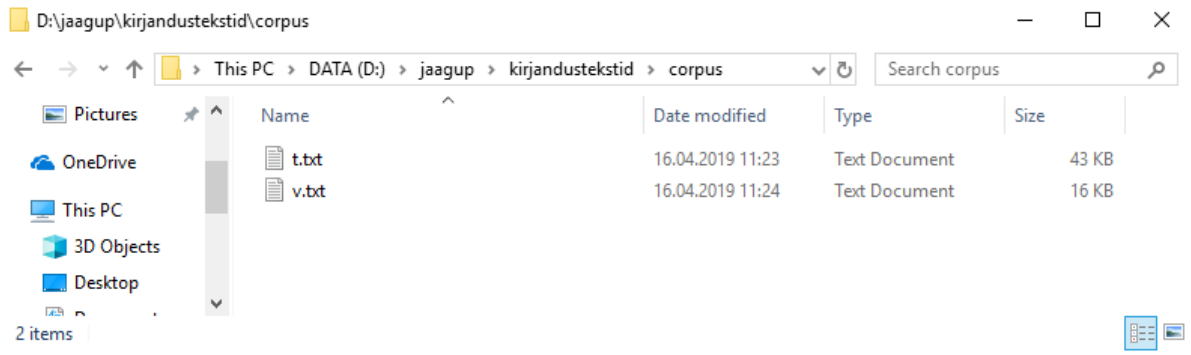
Kirjandustekstide võrdlus

Stilomeetria põhiline rakendusala ongi ilukirjanduslikud teosed ja autorite tuvastus. Siin sisendiks peatükk Tammsaare

```
https://et.wikisource.org/wiki/T%C3%B5de_ja_%C3%B5igus_I/XXXV
```

ning peatükk Vilde teosest

https://et.wikisource.org/wiki/Mahtra_s%C3%B5da/20



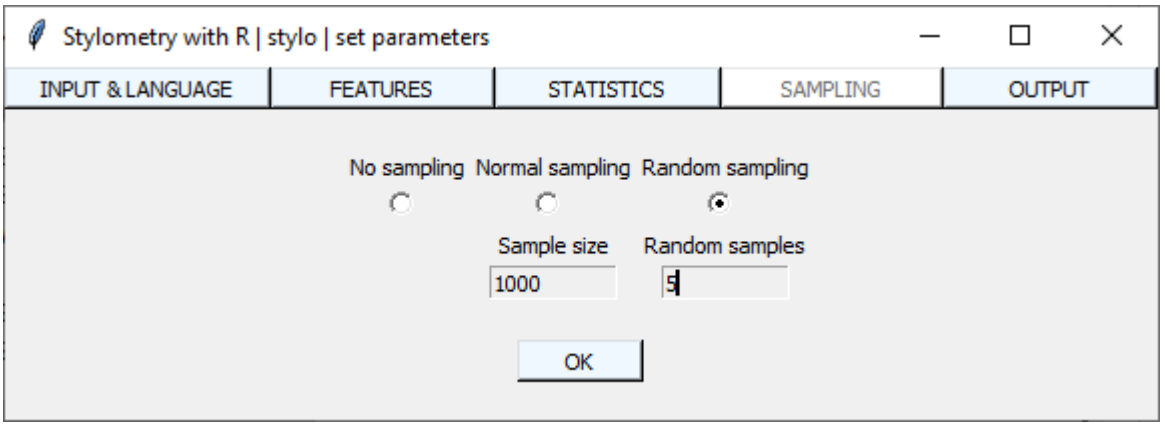
Kataloog paika

```
setwd("d:/jaagup/kirjandustekstid/")
```

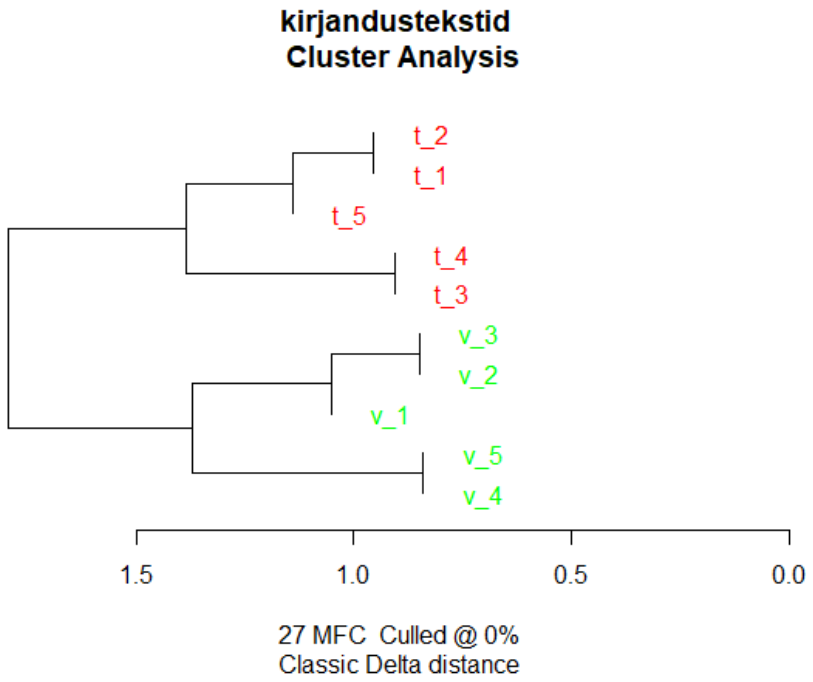
Programm käima

```
stylo()
```

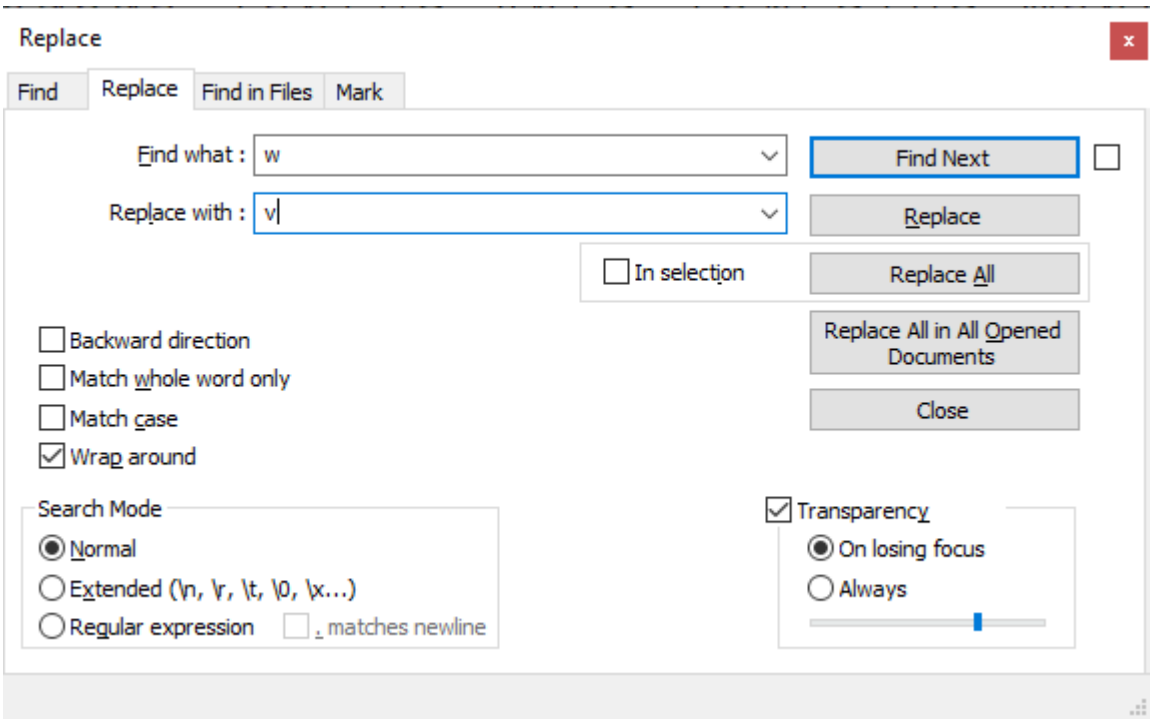
Pikemate tekstide võrdlemisel on põhjust võtta ports lõike iga teksti seest. Nii näeb, kuivõrd varieerub tekst ise ning kui suured on sellega võrreldes erinevused tekstide vahel



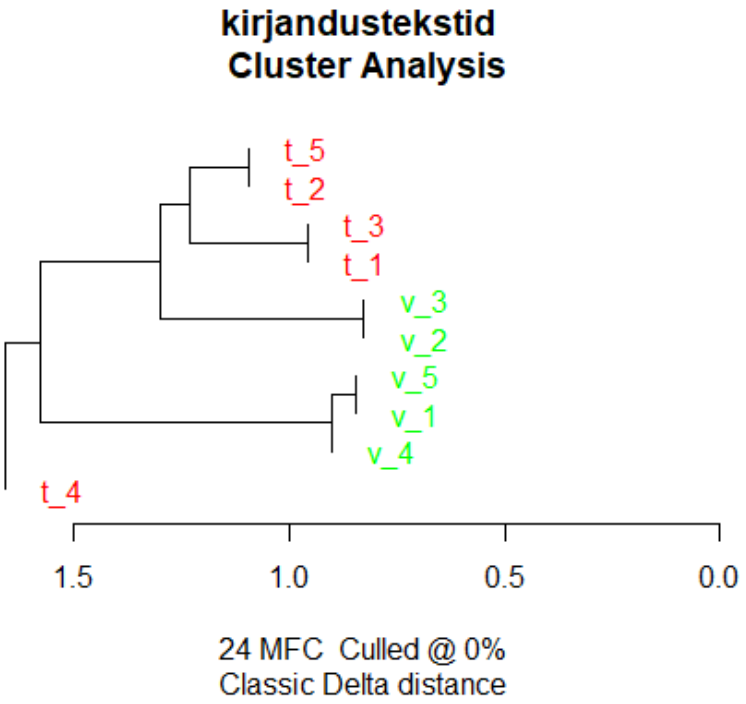
Pildi järgi paistab, et Tammsaare teksti kõik lõigud sattusid ühte alampuusse ning Vilde omdae teise



Esimese hooga tundub, et ju siis on programm nõnda hea aimaja. Tekste lähemalt uurides selgus aga, et Vilde kasutas lausetes w-tähte, Tammsaare aga v-tähte. Nõnda annab tähtede kaupa võrreldes märgatava erinevuse juba selle sümboli kasutus. Vahetame Vilde tekstis kaksisveed ühekordsete vastu



Uuesti läbi lastutena pole erinevused enam nõnda selged, aga mõningane grupeerumine siiski paistab.



Harjutus

- Lisa Tammsaare ja Vilde tekste. Nende romaanide peatükke leiab aadressilt `http://www.tlu.ee/~jaagup/andmed/keel/romaanid/`
- Rakenda uurimise juures mitmeid meetodeid, võrdle tulemusi.
- Lisa tekst mõnest oma kirjutisest ja jälgi, kuivõrd see millelegi eelnevale sarnaneb
- Otsi kirjanike tekstide kõrvale oma pikem kirjutis (või hulk lõike kokku väiksematest kirjutistest) ning lisa juurde ka pinginaabri koostatud pikem tekst. Koosta eri meetodite abil ülevaated, kuidas tekstid sisemiselt ning omavahel sarnanevad ja erinevad. Vormistage tulemused eraldi dokumendina kus näha nii rohkem kui vähem eristavate meetodite tulemused.

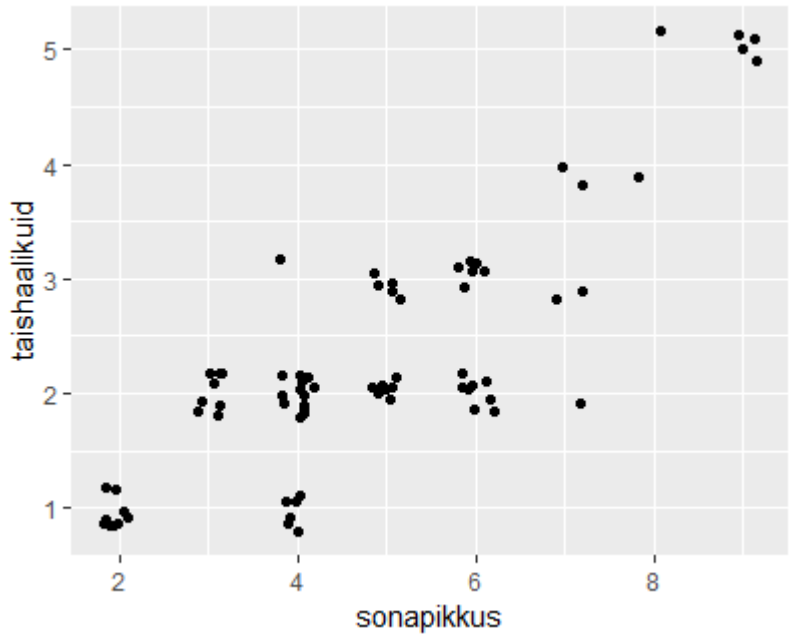
Regressioon

Lineaarne regressioon aitab ennustada arvulisi väärtusi vastavalt treeningandmetele. Sisendiks tuttavad sõnad

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

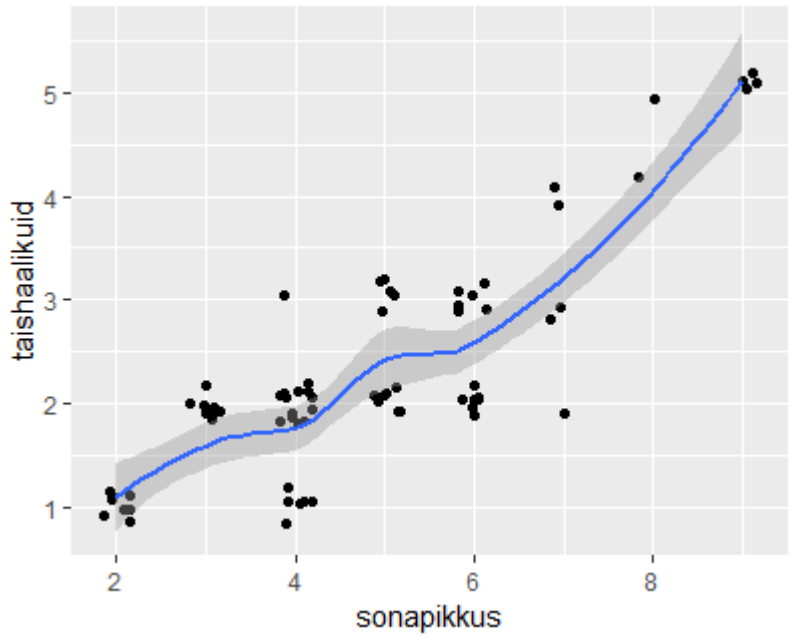
Joonis selle kohta, kuidas sõnade pikkus ning täishäälikute arv sõnas omavahel seotud on, geom_jitter'i parameetrid näitavad, kui suures vahemikus sisendandmeid "loksutatakse", et punktid üksteise peale ei jääks.

```
> sonad %>% filter(lugu=="kungla") %>% ggplot(aes(sonapikkus, taishaalikuid)) +
geom_jitter(width=0.2, height=0.2)
```



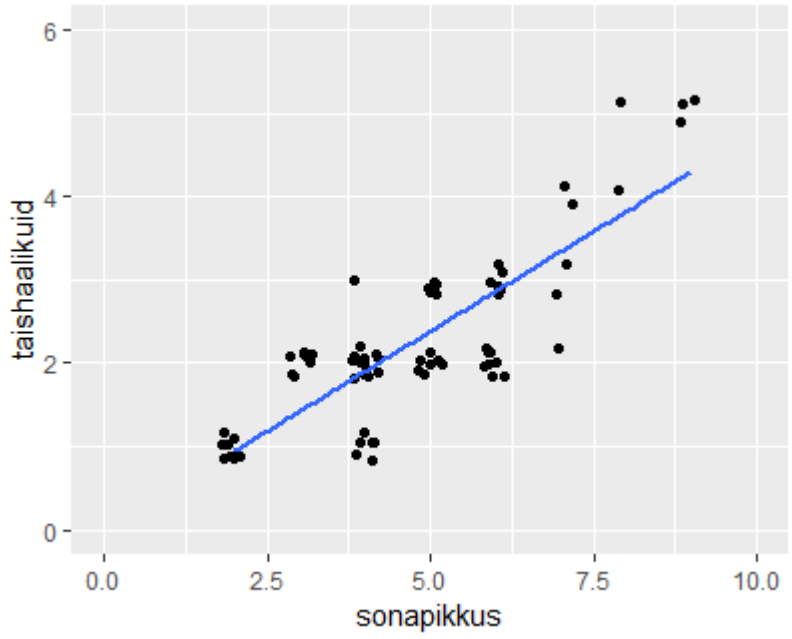
Seosejoone saab lisada parameetriga geom_smooth(). Vaikimisi võetakse algoritmiks paindlik kõverjoon. Hall ala näitab arvutuslikku standardviga, ehk piirkonda, kus väärtus 95% tõenäosusega usutav on

```
> sonad %>% filter(lugu=="kungla") %>% ggplot(aes(sonapikkus, taishaalikuid)) +
geom_jitter(width=0.2, height=0.2) + geom_smooth()
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Matemaatiliselt lihtsama sirgjoone leidmiseks tasub meetodiks määrata `lm` ehk linear modelling, `se=FALSE` ütleb, et standardvea (standard error) halli ala pole vaja lisada.

```
> sonad %>% filter(lugu=="kungla") %>% ggplot(aes(sonapikkus, taishaalikuid)) +
geom_jitter(width=0.2, height=0.2) + geom_smooth(method="lm", se = FALSE) + xlim(0, 10) +
ylim(0, 6)
```



Siit saab silma järgi vaadata, et veidi vähem kui viie tähe pikkustes sõnades on keskeltläbi kaks täishäälikut.

Arvulise lähenemise puhul tasub mudeli jaoks eraldi käsklus välja kutsuda. Saadakse vastus, et keskeltläbi on ilma täishäälikuteta sõna 1,45 tähe pikkune, iga lisanduv täishäälikutäht lisab sõna pikkusele 1,46 tähte.

```
> lm(sonapikkus~taishaalikuid, data=sonad %>% filter(lugu=="kungla"))

Call:
lm(formula = sonapikkus ~ taishaalikuid, data = sonad %>% filter(lugu ==
    "kungla"))

Coefficients:
  (Intercept)  taishaalikuid
         1.45           1.46
```

Lisades käsu `summary` on R veidi jutukam

Residuals näitab, et kui palju on millises suuruses möödaarvestusi. Sõnapikkus on näitandmete puhul ennustatust 1,83 tähe jagu lühem kuni 2,62 tähe jagu pikem. Pooltel juhtudel jääb arvutusviga -0,8 kuni +0,6 tähe piiresse.

Alla tabelisse lisandusid tõusu ja vabaliikme vahemikhinnangud. Ehk siis ilma täishäälikuteta sõna võiks olla keskeltläbi 1,45 +/- 0,27 tähe pikkune ning iga täishäälik võiks lisada 1,46 +/- 0,11 tähte. Tõenäosus, et vastav seos puuduks oleks esimesel juhul üks miljonist ning teisel kaks kümnest kvintiljonist (2e-16 ehk kaks korda kümme astmel miinus kuusteist)

```
> summary(lm(sonapikkus~taishaalikuid, data=sonad %>% filter(lugu=="kungla")))

Call:
lm(formula = sonapikkus ~ taishaalikuid, data = sonad %>% filter(lugu ==
    "kungla"))

Residuals:
    Min      1Q  Median      3Q     Max
-1.8309 -0.8309 -0.3706  0.6294  2.6294

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.4499     0.2788   5.201 1.74e-06 ***
taishaalikuid   1.4603     0.1118  13.058  < 2e-16 ***
```

Mudel tasub salvestada ning edasi võib juba mudeli järgi ennustada. Küsime ennustused sõnapikkustekohta sõnades, milles on 2, 3, 4 või viis täishäälikut. Saame vasted 4.370579 5.830909 7.291240 ja 8.751570

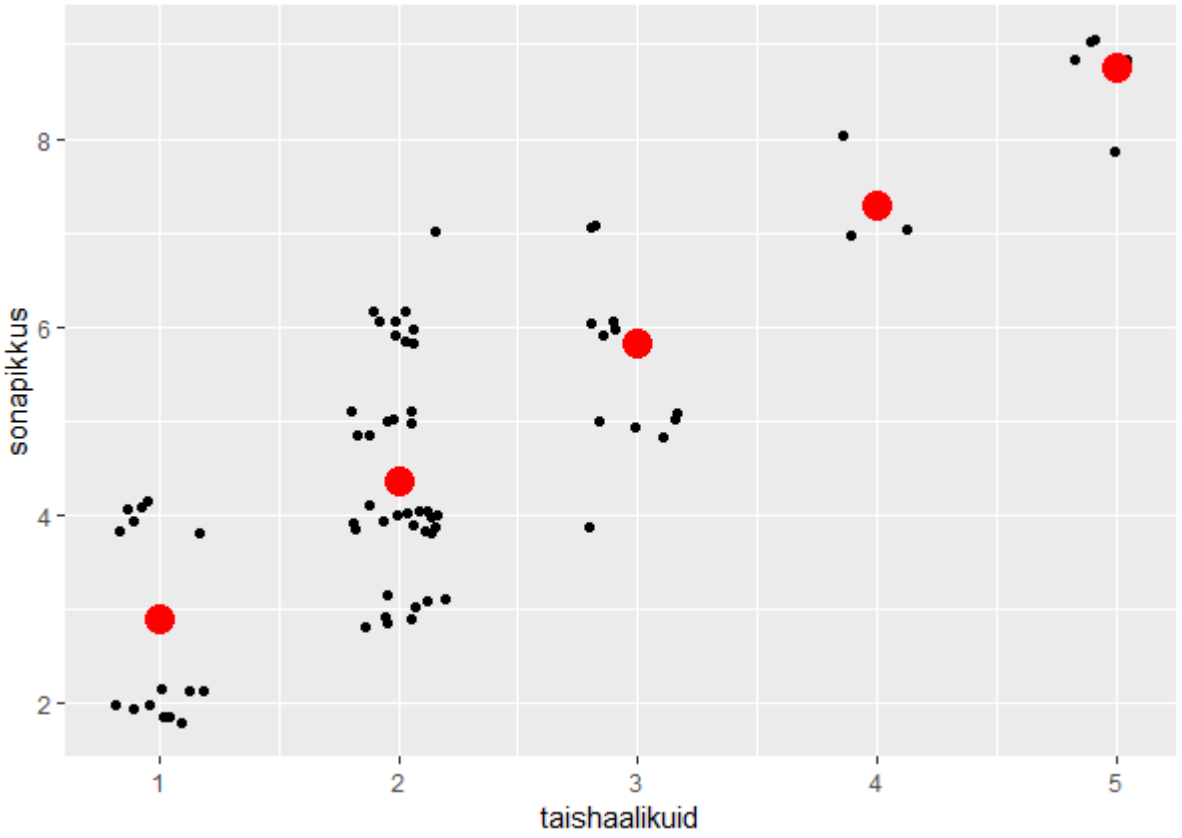
```
> mudel=lm(sonapikkus~taishaalikuid, data=sonad %>% filter(lugu=="kungla"))

> predict(mudel, tibble(taishaalikuid=c(2, 3, 4, 5)))
       1        2        3        4
4.370579 5.830909 7.291240 8.751570
```

Mugavamaks käskluseks paneme uuritavad täishäälikute arvud tibble-tüüpi tabelisse veeruna ning siis lisame teise veeruna ennustatud sõnapikkused.

```
> uuritav=tibble(taishaalikuid=c(1, 2, 3, 4, 5))
> uuritav$sonapikkus=predict(mudel, uuritav)
> uuritav
# A tibble: 5 x 2
  taishaalikuid sonapikkus
          <dbl>      <dbl>
1             1       2.91
2             2       4.37
3             3       5.83
4             4       7.29
5             5       8.75
```

Edasi algsed andmed koos ennustustega joonisele.

```
> sonad %>% filter(lugu=="kungla") %>% ggplot(aes(taishaalikuid, sonapikkus)) +
geom_jitter(width=0.2, height=0.2) + geom_point(data=uuritav, color="red", size=5)
```



Harjutus

- Tehke näide läbi
- Leidke regressioonikordaja ja vabaliige sõnapikkuse ja täishäälikute arvu seoses Lambipirni jutu sõnades
  `http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt`
- Ennustage Lambipirni jutu sõnade keskmist tähtede arvu kahe ning viie täishäälikuga sõnas.

- Kuvage tulemus joonisena.

Mitme parameetriga mudel

Sõna tähtede arv sõltub mingil moel arvatavasti lisaks täishäälikute arvule ka sulghäälikute arvust. Kuidas täpsemalt, sellele aitab vastuse saada loodud mudel. Kui vaid täishäälikuid arvestades näitas alumine kvartiil 0,83 vähem, siis koos sulghäälikutega 0,55. Samuti maksimumviga läks mõnevõrra väiksemaks. Täishääliku mõju jäi samasuguseks, iga lisanduv sulghäälik suurendab sõna pikkust keskmiselt 0,76 tähe jagu

```
> mudel=lm(sonapikkus~taishaalikuid+sulghaalikuid, data=sonad %>% filter(lugu=="kungla"))
> summary(mudel)

Call:
lm(formula = sonapikkus ~ taishaalikuid + sulghaalikuid, data = sonad %>%
    filter(lugu == "kungla"))

Residuals:
     Min       1Q   Median       3Q      Max
-1.61460 -0.55886 -0.07116  0.62599  2.14482

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.9421     0.2412   3.905  0.00021 ***
taishaalikuid   1.4566     0.0910  16.006  < 2e-16 ***
sulghaalikuid   0.7594     0.1228   6.186 3.37e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Ennustus kahe tunnuse põhjal:

```
uuritav=tibble(taishaalikuid=c(2, 2, 5, 5), sulghaalikuid=c(1, 4, 1, 4))

> uuritav
# A tibble: 4 x 2
  taishaalikuid sulghaalikuid
          <dbl>         <dbl>
1             2             1
2             2             4
3             5             1
4             5             4
```

Ülalt vaadatavate koefitsientide põhjal arvutus, kuidas leida kahe täishääliku ja ühe sulghäälikuga sõna keskmist pikkust. Sõna eeldatav pikkus, kui täis- ja sulghäälikud puuduvad, on mudeli tabeli järgi 0,9421. Iga täishäälik lisab pikkust 1,4566 tähe jagu - praegu neid kaks tükki. Sulghäälik lisab keskmiselt veel 0.7594 tähepikkust. Nii tulebki esimesele sõnale ennustatav pikkus 4,6147 ehk ümardatult 4,61.

```
> uuritav$sonapikkus=predict(mudel, uuritav)
> uuritav
# A tibble: 4 x 3
  taishaalikuid sulghaalikuid sonapikkus
```

```
          <dbl>          <dbl>      <dbl>
1             2              1       4.61
2             2              4       6.89
3             5              1       8.98
4             5              4      11.3
```

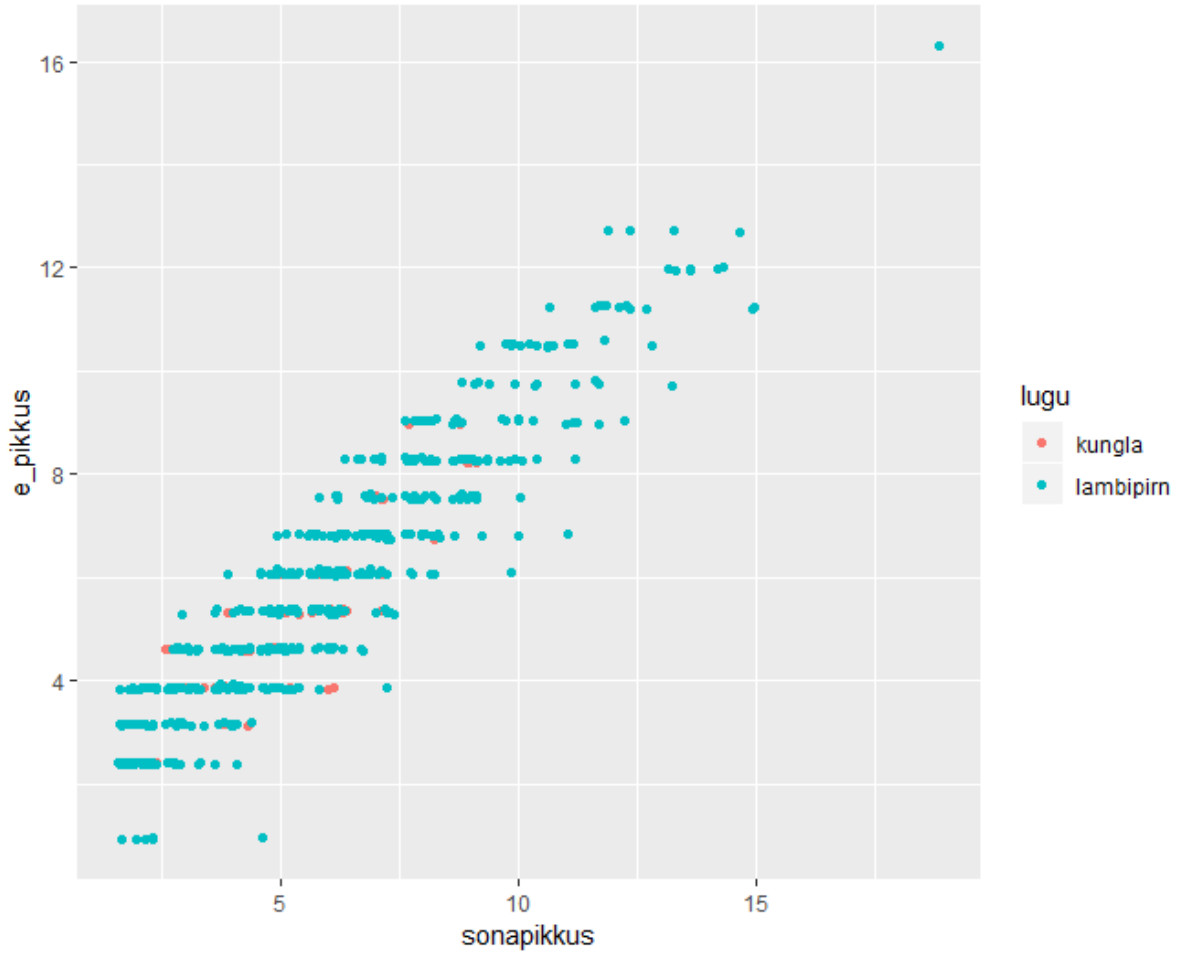
```
> 0.9421+2*1.4566+1*0.7594
[1] 4.6147
```

Lisame sõnade tabelile ennustatava pikkuse

```
> sonad$e_pikkus=predict(mudel, sonad)
> head(sonad)
# A tibble: 6 x 6
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid e_pikkus
  <chr>  <chr>        <int>         <int>         <int>    <dbl>
1 kungla kui              3             2             1     4.61
2 kungla kungla           6             2             2     5.37
3 kungla rahvas           6             2             0     3.86
4 kungla kuldsel          7             2             2     5.37
5 kungla aal              3             2             0     3.86
6 kungla kord             4             1             2     3.92
```

Jooniselt vaatame, millise loo kui pikad sõnad kui täpselt ennustati. Ühel teljel tegelik pikkus ning teisel ennustatu.

```
> sonad %>% ggplot(aes(sonapikkus, e_pikkus, color=lugu)) + geom_jitter()
```



Oma panuse ennustusse võib anda ka rühmatunnus. Siin mudel, kus sõna pikkust püütakse leida vastavalt sellele, mitu täishäälikut on sõnas ning millise looga on tegemist. Väljundist paistab, et kui tegemist on lambipirni-jutuga Kungla rahva laulu asemel, siis on sõna eeldatav keskmine pikkus kolmandiku tähe jagu suurem.

```
> lm(sonapikkus~taishaalikuid+lugu, sonad)

Call:
lm(formula = sonapikkus ~ taishaalikuid + lugu, data = sonad)

Coefficients:
  (Intercept)  taishaalikuid  lugulambipirn
       0.2371         1.9954         0.3313
```

Sama vastus ka ennustuse juures. Kahe täishäälikuga sõna lambipirni-jutus eeldatava keskmise pikkusega 4,56, Kungla rahva loos 4,23

```
> mudel=lm(sonapikkus~taishaalikuid+lugu, sonad)
> uuritav=tibble(taishaalikuid=c(2, 2, 5, 5), lugu=c("kungla", "lambipirn", "kungla",
"lambipirn"))
> uuritav$sonapikkus=predict(mudel, uuritav)
> uuritav
# A tibble: 4 x 3
```

```
  taishaalikuid lugu      sonapikkus
          <dbl> <chr>          <dbl>
1             2 kungla          4.23
2             2 lambipirn       4.56
3             5 kungla         10.2
4             5 lambipirn      10.5
```

Harjutus

- Tehke näited läbi
- Ennustage regressiooni abil täishäälikute arv sõnas sõnapikkuse järgi (soovi korral Kungla rahva loos)
- Illustreerige tulemust joonisega
- Koosta arvutuskäik viietähelise sõna täishäälikute arvu ennustamiseks
- Ennustage regressiooni abil täishäälikute arv sõnas sõnapikkuse ning sulghäälikute arvu järgi
- Too välja koefitsiendid kummagi tunnuse kohta, koosta arvutuskäik täishäälikute arvu ennustamiseks viietähelise kahe sulghäälikuga sõna näitel
- Ennustage täishäälikute arv sõnas vastavalt sõnapikkusele, sulghäälikute arvule ning loo nimetusele

```
> lm(taishaalikuid~sonapikkus+sulghaalikuid, data=sonad %>% filter(lugu=="kungla"))

Call:
lm(formula = taishaalikuid ~ sonapikkus + sulghaalikuid, data = sonad %>%
    filter(lugu == "kungla"))

Coefficients:
  (Intercept)     sonapikkus  sulghaalikuid
    -0.008958       0.535932      -0.405016

> -0.009 + 5*0.536 -2*0.41
[1] 1.851
```

Klasterdamine

ehk rühmadeks jagamine

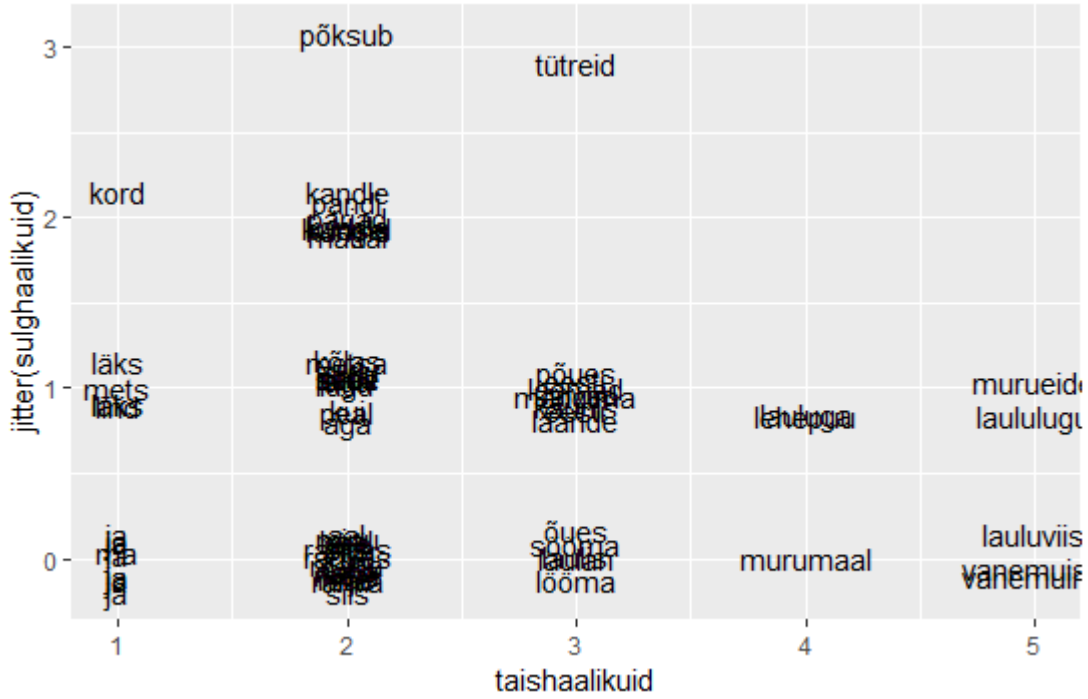
Näite sisendiks tuttavad sõnad

```
sonad=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_haalikud.txt")
```

Sõnade paiknemine vastavalt täis- ja sulghäälikute arvule

```
sonad %>% filter(lugu=="kungla") %>%
```

```
ggplot(aes(taishaalikuid, jitter(sulghaalikuid), label=sona)) + geom_text()
```



Käsklus kmeans jagab sõnad loetelus soovitud arvuks rühmadeks. Praeguses näites palutakse sõnad koondada kahe keskme ümber.

```
> kunglasonad=sonad %>% filter(lugu=="kungla")
> ryhmad=kmeans(kunglasonad %>% select(taishaalikuid, sulghaalikuid), centers=2)
```

Vastuse muutujas cluster on järjekorras igale sõnale pakutud rühma number

```
> ryhmad$cluster
 [1] 2 2 1 2 1 2 2 1 1 1 1 1 2 2 2 1 2 2 2 1 2 2 1 1 2 1 2 2 1 2 1 1 1 1 2 2 1 1 2 1 1 2 1 2 1 1 2 2
[49] 2 2 2 1 2 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 2 1 1 2 1 1
```

Muutujast centers leiab rühmade keskkohad

```
> ryhmad$centers
  taishaalikuid sulghaalikuid
1      2.871795     0.3076923
2      1.611111     1.0833333
```

Kinnitame rühmade numbrid sõnade külge

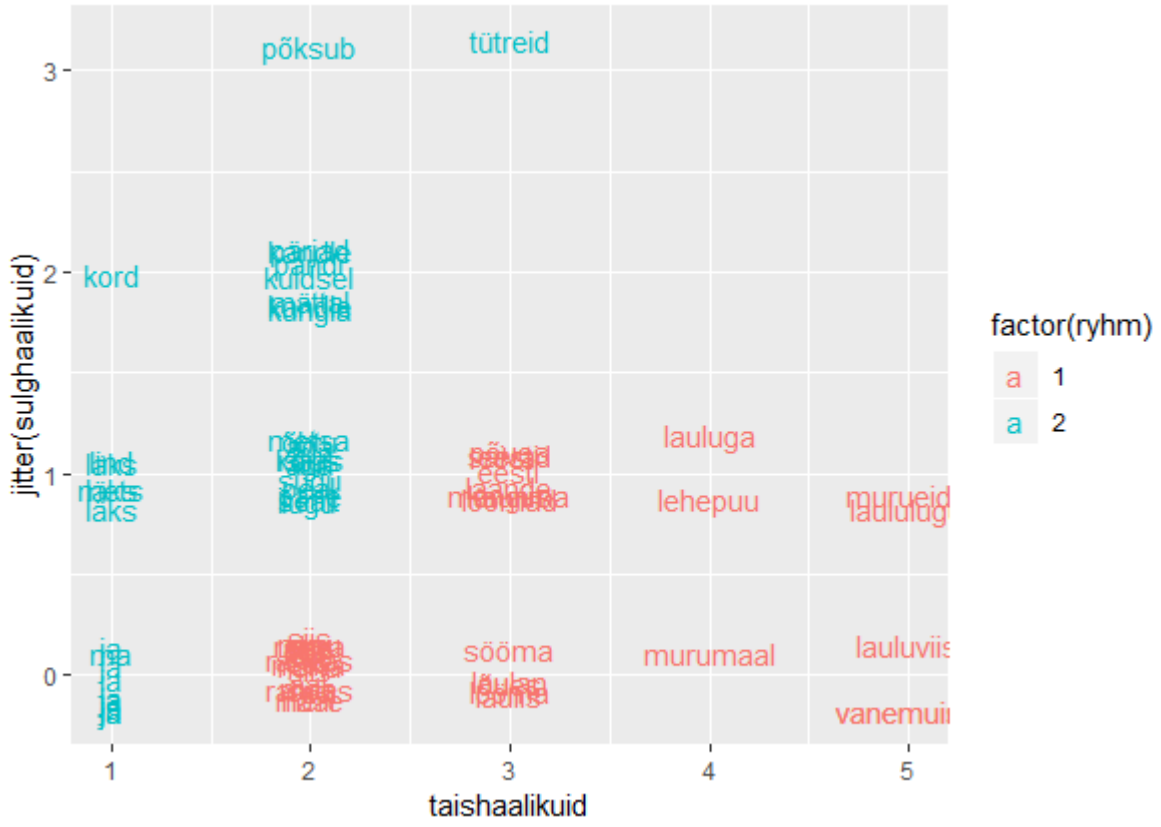
```
> kunglasonad$ryhm=ryhmad$cluster
```

```
> head(kunglasonad)
# A tibble: 6 x 6
  lugu   sona    sonapikkus taishaalikuid sulghaalikuid  ryhm
  <chr>  <chr>        <int>         <int>         <int> <int>
1 kungla kui              3             2             1     2
2 kungla kungla           6             2             2     2
```

```
3 kungla rahvas            6             2             0     1
4 kungla kuldsel           7             2             2     2
5 kungla aal               3             2             0     1
6 kungla kord              4             1             2     2
```

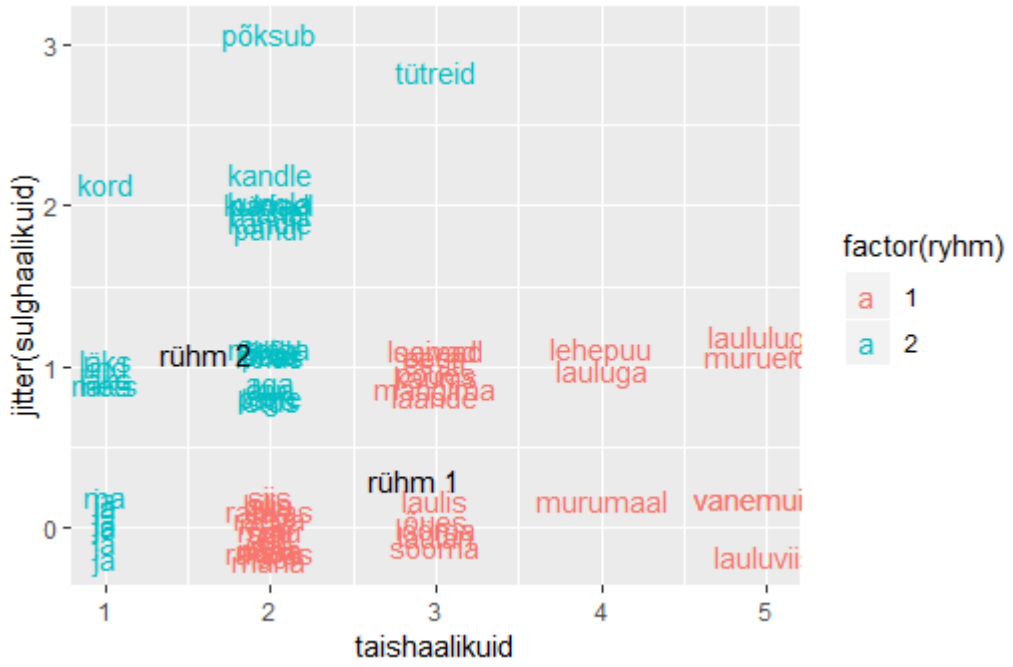
ja vaatame joonisel, et kuidas sõnad ja rühmad kaheteljelisel skaalal paiknevad

```
 kunglasonad %>% ggplot(aes(taishaalikuid, jitter(sulghaalikuid), color=factor(ryhm), label=sona)) + geom_text()
```



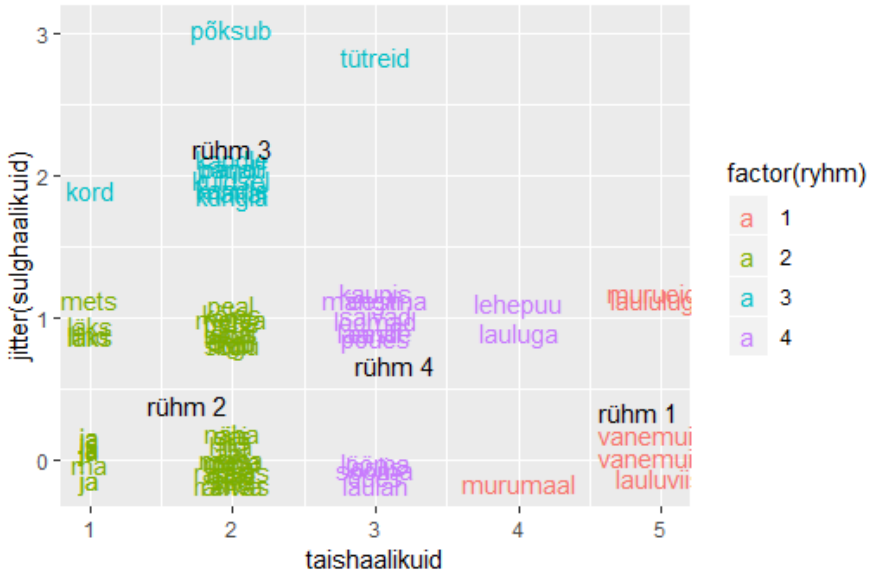
Rühmade keskmed ka juurde

```
ggplot() +
  geom_text(aes(taishaalikuid, jitter(sulghaalikuid), color=factor(ryhm), label=sona), data=kunglasonad)+
  geom_text(aes(taishaalikuid, sulghaalikuid, label=nr), data=as_tibble(ryhmad$centers) %>% mutate(nr=paste("rühm ", row_number(), sep="")))
```



Sama arvutus, aga rühmade arvuks neli

```
ryhmad=kmeans(kunglasonad %>% select(taishaalikuid, sulghaalikuid), centers=4)
kunglasonad$ryhm=ryhmad$cluster
ggplot() +
  geom_text(aes(taishaalikuid, jitter(sulghaalikuid), color=factor(ryhm), label=sona),
data=kunglasonad)+
  geom_text(aes(taishaalikuid, sulghaalikuid, label=nr), data=as_tibble(ryhmad$centers) %>%
mutate(nr=paste("rühm ", row_number(), sep="")))
```



Harjutus

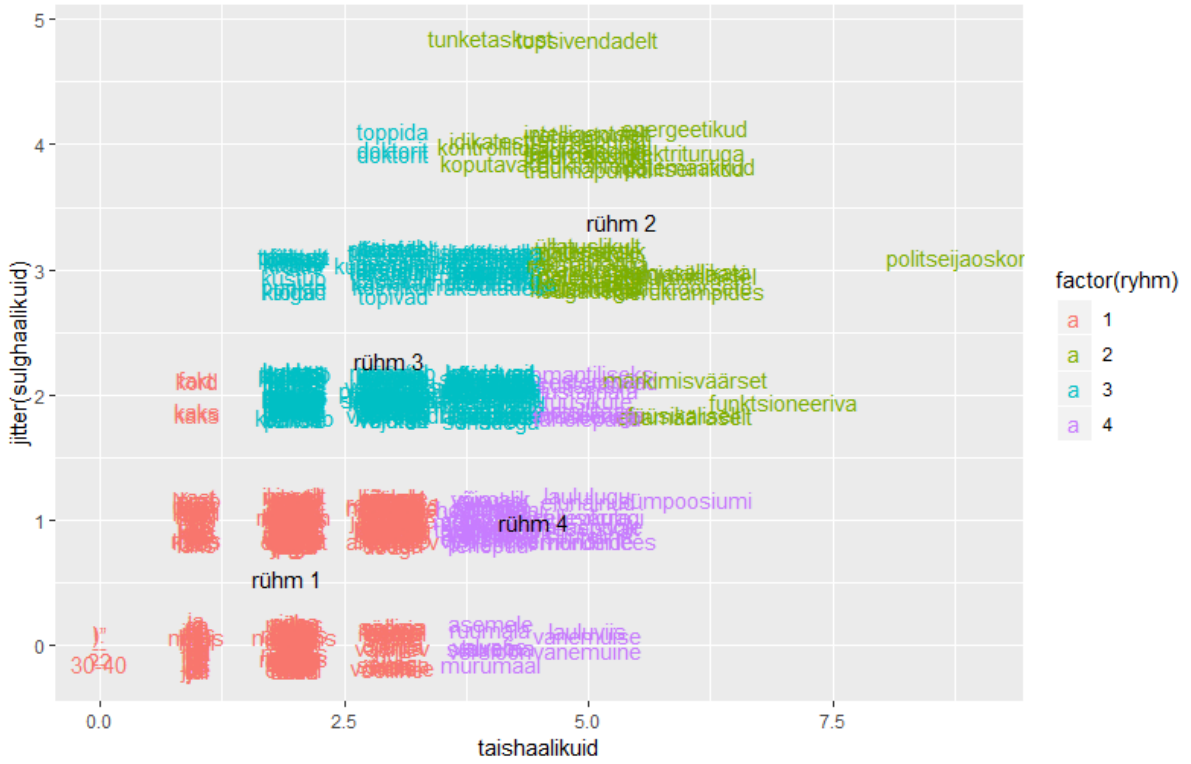
- Pange näited tööle
- Jagage Kungla rahva sõnad täis- ja sulghäälikute arvu järgi kolme rühma, koostage joonis
- Jagage Lambipirni jutu sõnad täis- ja sulghäälikute arvu järgi kahte, kolme ja nelja rühma, vaadake, kas ja millisel puhul oleks rühmi kõige selgem iseloomustada
- Jagage kogu sõnadetabeli sõnad samade tunnuste järgi nelja rühma. Näidake kummagi loo puhul, mitu protsenti sõnadest millisesse rühma sattus.

```
ryhmad=kmeans(sonad %>% select(taishaalikuid, sulghaalikuid), centers=4)
sonad$ryhm=ryhmad$cluster
sample_n(sonad, 10)
```

```
# A tibble: 10 x 6
   lugu   sona    sonapikkus taishaalikuid sulghaalikuid  ryhm
   <chr>  <chr>        <int>         <int>         <int> <int>
 1 lambi~ ukse             4             2             1     1
 2 lambi~ poole            5             3             1     1
 3 lambi~ kirurg           6             2             2     3
 4 lambi~ sealt            5             2             1     1
 5 lambi~ saavad           6             3             1     1
 6 lambi~ !”               2             0             0     1
 7 lambi~ aga              3             2             1     1
 8 lambi~ naeruk~         14             6             3     2
 9 lambi~ ühes             4             2             0     1
10 kungla kaunis           6             3             1     1
```

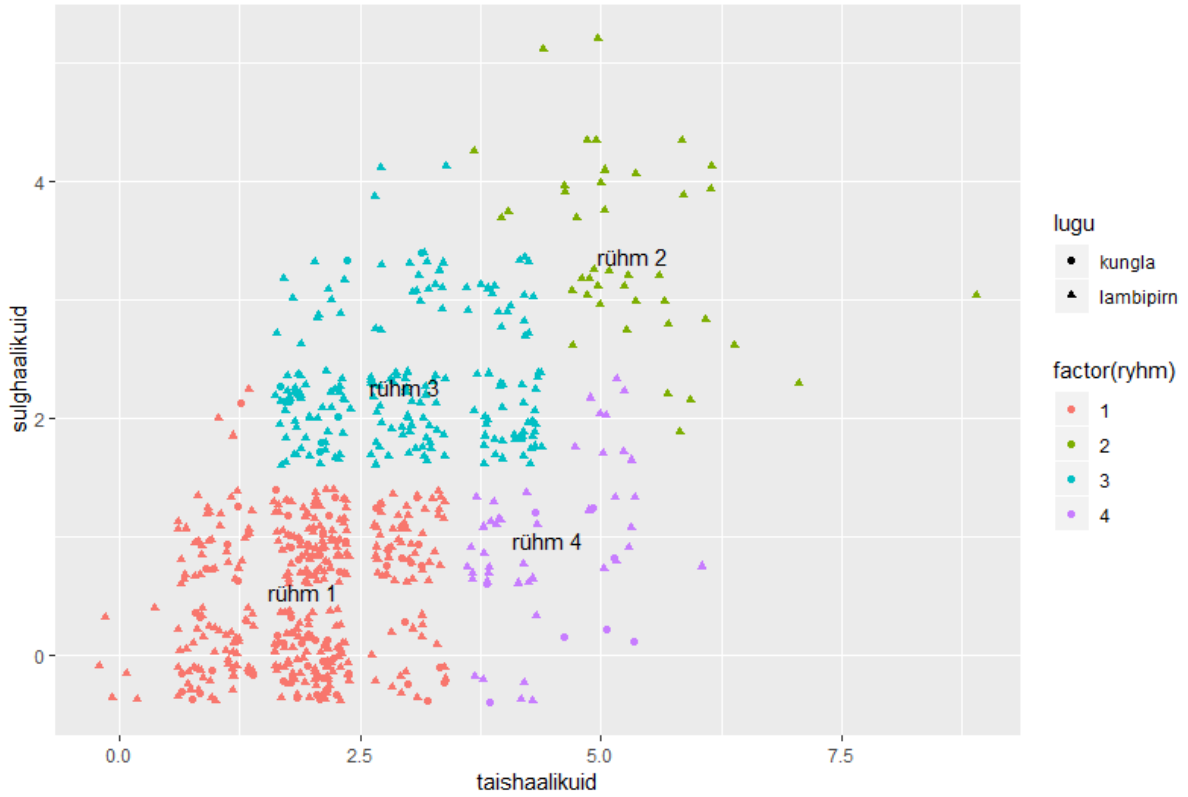
Sõnade jaotumine rühmiti

```
ggplot() +
  geom_text(aes(taishaalikuid, jitter(sulghaalikuid), color=factor(ryhm), label=sona),
data=sonad)+
  geom_text(aes(taishaalikuid, sulghaalikuid, label=nr), data=as_tibble(ryhmad$centers) %>%
mutate(nr=paste("rühm ", row_number(), sep="")))
```



Lisaks loo järgi punkti kuju

```
ggplot() +
  geom_jitter(aes(taishaalikuid, sulghaalikuid, color=factor(ryhm), shape=lugu),
data=sonad)+
  geom_text(aes(taishaalikuid, sulghaalikuid, label=nr), data=as_tibble(ryhmad$centers) %>%
mutate(nr=paste("rühm ", row_number(), sep="")))
```



Sõnade arv rühmiti

```
> sonad %>% group_by(lugu, ryhm) %>% summarise(kogus=n())
# A tibble: 7 x 3
# Groups:   lugu [?]
  lugu       ryhm kogus
  <chr>     <int> <int>
1 kungla        1    58
2 kungla        3     9
3 kungla        4     8
4 lambipirn     1   327
5 lambipirn     2    42
6 lambipirn     3   184
7 lambipirn     4    44
```

juurde iga loo sõnade arv

```
sonad %>% group_by(lugu, ryhm) %>% summarise(kogus=n()) %>%
  group_by(lugu) %>% mutate(sonuloos=sum(kogus))

# A tibble: 7 x 4
# Groups:   lugu [2]
  lugu       ryhm kogus sonuloos
  <chr>     <int> <int>    <int>
1 kungla        1    58       75
2 kungla        3     9       75
3 kungla        4     8       75
4 lambipirn     1   327      597
5 lambipirn     2    42      597
6 lambipirn     3   184      597
7 lambipirn     4    44      597
```

Milline osa vastava loo sõnadest on konkreetses rühmas

```
sonad %>% group_by(lugu, ryhm) %>% summarise(kogus=n()) %>%
  group_by(lugu) %>% mutate(sonuloos=sum(kogus), protsent=100*kogus/sonuloos) %>% ungroup()
```

```
# A tibble: 7 x 5
  lugu       ryhm kogus sonuloos protsent
  <chr>     <int> <int>    <int>    <dbl>
1 kungla        1    58       75    77.3
2 kungla        3     9       75    12
3 kungla        4     8       75    10.7
4 lambipirn     1   327      597    54.8
5 lambipirn     2    42      597     7.04
6 lambipirn     3   184      597    30.8
7 lambipirn     4    44      597     7.37
```

Samad arvud laia tabelisse

```
paiknemised <- sonad %>% group_by(lugu, ryhm) %>% summarise(kogus=n()) %>%
  group_by(lugu) %>% mutate(sonuloos=sum(kogus), protsent=100*kogus/sonuloos) %>%
  ungroup() %>% select(lugu, ryhm, protsent) %>% spread(lugu, protsent, fill=0)

paiknemised
# A tibble: 4 x 3
   ryhm kungla lambipirn
  <int>  <dbl>     <dbl>
1     1   77.3     54.8
2     2    0        7.04
3     3   12       30.8
4     4   10.7      7.37
```

Rühmade keskkohad

```
> ryhmad$centers
  taishaalikuid sulghaalikuid
1      1.909091     0.5376623
2      5.333333     3.3809524
3      2.963731     2.2746114
4      4.442308     0.9807692
```

Rühma number eraldi tulbaks

```
> ryhmad$centers %>% as_tibble() %>% mutate(ryhm=row_number())
# A tibble: 4 x 3
  taishaalikuid sulghaalikuid  ryhm
          <dbl>         <dbl> <int>
1          1.91         0.538     1
2          5.33         3.38      2
3          2.96         2.27      3
4          4.44         0.981     4
```

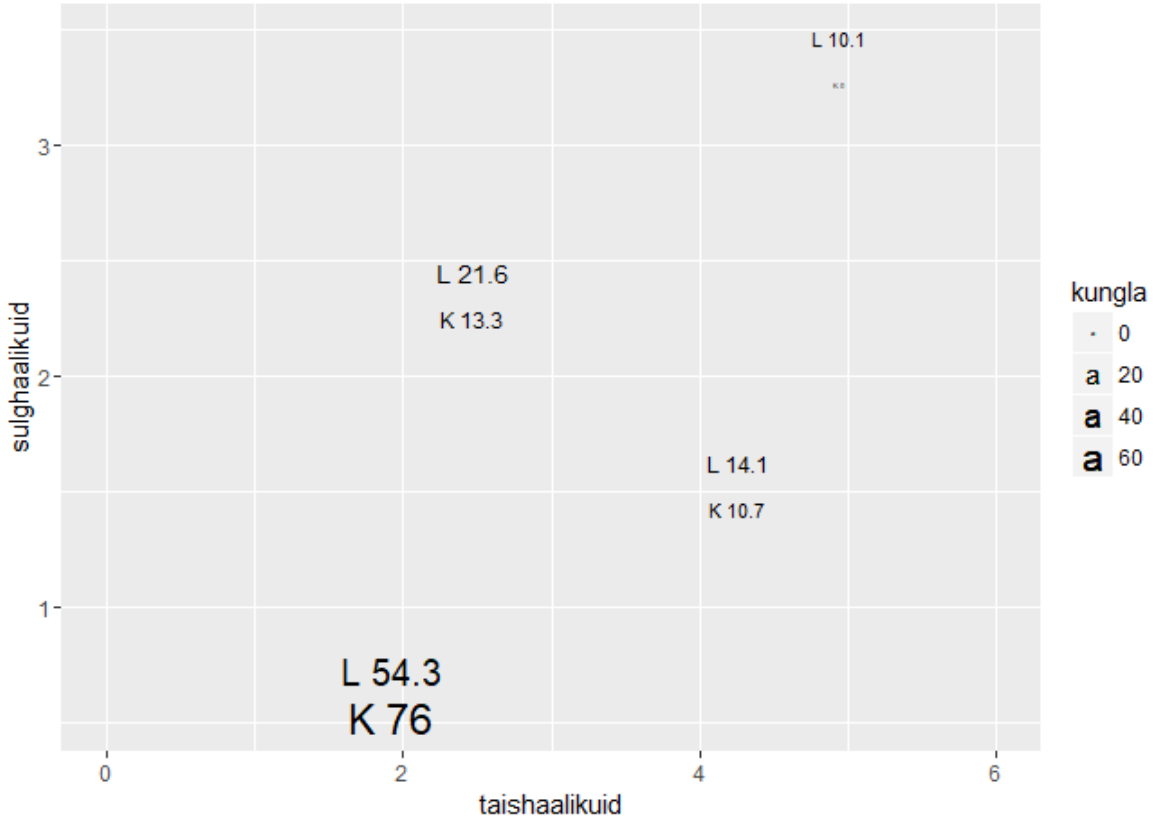
Juurde rühmakeskuste asukohad

```
paiknemised %>% inner_join(ryhmad$centers %>% as_tibble() %>% mutate(ryhm=row_number()))
```

```
Joining, by = "ryhm"
# A tibble: 4 x 5
   ryhm kungla lambipirn taishaalikuid sulghaalikuid
  <int>  <dbl>     <dbl>         <dbl>         <dbl>
1     1   77.3     54.8           1.91         0.538
2     2    0        7.04          5.33         3.38
3     3   12       30.8           2.96         2.27
4     4   10.7      7.37          4.44         0.981
```

koos joonisega

```
paiknemised %>% inner_join(ryhmad$centers %>% as_tibble() %>% mutate(ryhm=row_number()))
%>% ggplot(aes(taishaalikuid, sulghaalikuid)) + geom_text(aes(label=paste("K",
round(kungla, 1)), size=kungla)) + geom_text(aes(label=paste("L", round(lambipirn, 1)),
size=lambipirn, y=sulghaalikuid+0.2)) + xlim(0, 6)
```



Palju tunnuseid

Tasandile mahtuvate tunnuste puhul kannatab ka silmaga vaadata, et mis rohkem omavahel kokku puutuvad. K-keskmiste arvutamine aitab lihtsalt matemaatiliselt paremini näidata, et kuidas rühmad kujuneda võiksid - kusjuures ka seal on hea tahtmise korral võimalik mitme algoritmi vahel valida.

Näite sisendiks Eesti Vahekeele Korpuse tesktide mitmesugused andmed

```
tekstiandmed=read_csv("http://www.tlu.ee/~jaagup/andmed/keel/korpus/dokkoik.txt")
 colnames(tekstiandmed)
 [1] "kood"                    "korpus"                  "tekstikeel"
 [4] "tekstityyp"              "elukoht"                 "taust"
 [7] "vanus"                   "sugu"                    "emakeel"
[10] "kodukeel"                "keeletase"               "haridus"
[13] "abivahendid"             "A"                       "C"
[16] "D"                       "G"                       "H"
[19] "I"                       "J"                       "K"
[22] "N"                       "P"                       "S"
[25] "U"                       "V"                       "X"
[28] "Y"                       "Z"                       "kokku"
[31] "tahti"                   "sonu"                    "lauseid"
[34] "vigu"                    "veatyype"                "kolmetahelistepr"
[37] "viietahelistepr"         "kymnejarohkemtahelistepr" "kahesonalistepr"
[40] "kolmesonalistepr"        "kuuekuni9sonalistepr"    "kymnekuni20sonalistepr"
```

Kasutame sealt esimese hooga sõnaliikide sagedusi (tulbad A kuni Z)

```
> tekstiandmed %>% select(A:Z)
# A tibble: 12,724 x 16
       A     C     D     G     H     I     J     K     N     P     S     U     V     X
Y
   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
<int>
 1    25     0    14     0     3     0    19     5     3    17    54     0    35     0
0
 2     4     0     5     0     4     0    12     1     3    14    31     0    22     0
0
 3     9     0     6     0     2     0    13     1     3    17    53     0    25     0
2
```

Arvutame igale tekstile juurde neile pakutava rühma, esialgu jagame andmestiku neljaks rühmaks

```
> tekstiandmed$ryhm=kmeans(tekstiandmed %>% select(A:Z), centers=4)$cluster
```

Näitena välja esimeste tekstide keeletase, omadussõnade (Adjektiiv) ning nimisõnade (Substantiiv) arv ja algoritmi pakutud rühm

```
> tekstiandmed %>% select(keeletase, A, S, ryhm)
# A tibble: 12,724 x 4
   keeletase     A     S  ryhm
   <chr>     <int> <int> <int>
 1 B            25    54     3
 2 B             4    31     3
 3 B             9    53     3
 4 A            46   183     4
 5 B            43   182     4
 6 A            45   180     4
 7 A            44   173     4
 8 C           317  2171     1
 9 NA            0     0     3
10 C1           18    69     4
```

Märkus: vastus võib samade sisendandmete puhul tulla erinev, sest kmeans-käsklus kasutab rühmade esialgsete keskmiste leidmisel juhuslikke arve

Uurime, kuidas tekstide keeletasemed (mida sõnaliikide sageduste järgi rühmitamisel teada ei olnud) sattusid rühmadesse

```
tekstiandmed %>% group_by(ryhm, keeletase) %>% summarise(kogus=n())
 ryhm keeletase kogus
   <int> <chr>     <int>
 1     1 B             2
 2     1 C             7
 3     1 C1            2
 4     1 NA            1
 5     2 A            12
 6     2 B            24
 7     2 B1            2
 8     2 B2            5
 9     2 C            30
10     2 C1           20
# ... with 21 more rows
```

Edasi muudame andmestiku spread-käskluse abil laiale kujule, kus iga keeletase moodustab omaette tulba ning algoritmi pakutud rühm rea

```
ryhmatabel=tekstiandmed %>% group_by(ryhm, keeletase) %>% summarise(kogus=n()) %>%
spread(keeletase, kogus, fill=0) %>% ungroup()

 ryhmatabel
# A tibble: 4 x 11
   ryhm     A    A1    A2     B    B1    B2     C    C1    C2 `<NA>`
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
1     1     0     0     0     2     0     0     7     2     0      1
2     2    12     0     0    24     2     5    30    20    31    313
3     3  1187     1   213   828   349    87   184    14    59   6598
4     4   179     0     4   300    61   136   138    94    95   1748
```

Uurimiseks jätame alles vaid teadaoleva keeletasemega rühmad

```
> ryhmatabel=ryhmatabel %>% select(A:C2)
> ryhmatabel
# A tibble: 4 x 9
      A    A1    A2     B    B1    B2     C    C1    C2
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     0     0     0     2     0     0     7     2     0
2    12     0     0    24     2     5    30    20    31
3  1187     1   213   828   349    87   184    14    59
4   179     0     4   300    61   136   138    94    95
```

Tabelile tehtud hii-ruut test näitab, et keeletasemete erinevused rühmiti on tugevalt üldistatavad.

```
> ryhmatabel  %>% chisq.test()

        Pearson's Chi-squared test

data:  .
```

```
X-squared = 991.34, df = 24, p-value < 2.2e-16
```

Arvutatud andmestikus mõningane ülevaade.

Tekstide arv keeletasemete kaupa

```
> ryhmatabel %>% colSums()
   A   A1   A2    B   B1   B2    C   C1   C2
1378    1  217 1154  412  228  359  130  185
```

Tekstide arv rühmade kaupa

```
> ryhmatabel %>% rowSums()
[1]   11  124 2922 1007
```

Keeletasemega tekstide üldarv

```
> sum(ryhmatabel)
[1] 4064
```

Näitarvutus, et kui palju võiks olla A keeletasemega tekste neljandas rühmas, kui teksti sattumine rühma ei sõltuks keeletasemest. A tasemega tekstide üldarv on 1378, neljanda rühma tekstide arv 1007. Neljandasse sattumise tõenäosus on igal tekstil nõnda 1007/1064 ehk ca 0,25. 1378st A-tekstist võiks nõnda 4. rühma sattuda nõnda 341 teksti.

```
> a4=1378*(1007/4064)
> a4
[1] 341.4483
```

Rakendame seda arvutust sapply abil kõigile rühmadele

```
> sapply(1:nrow(ryhmatabel),
function(ryhmanr){colSums(ryhmatabel)*rowSums(ryhmatabel)[ryhmanr]})/sum(ryhmatabel)
          [,1]        [,2]        [,3]        [,4]
A  3.729822835 42.04527559 990.7765748 341.4483268
A1 0.002706693  0.03051181   0.7189961   0.2477854
A2 0.587352362  6.62106299 156.0221457  53.7694390
B  3.123523622 35.21062992 829.7214567 285.9443898
B1 1.115157480 12.57086614 296.2263780 102.0875984
B2 0.617125984  6.95669291 163.9311024  56.4950787
C  0.971702756 10.95374016 258.1195866  88.9549705
C1 0.351870079  3.96653543  93.4694882  32.2121063
C2 0.500738189  5.64468504 133.0142717  45.8403051
```

pöörame tagasi ning mugavamaks vaatamiseks ümardame väärtused

```
teoreetiline=t(sapply(1:nrow(ryhmatabel),
function(ryhmanr){colSums(ryhmatabel)*rowSums(ryhmatabel)[ryhmanr]})/sum(ryhmatabel))
> teoreetiline %>% round(2)
          A   A1     A2      B     B1     B2      C    C1     C2
[1,]   3.73 0.00   0.59   3.12   1.12   0.62   0.97  0.35   0.50
[2,]  42.05 0.03   6.62  35.21  12.57   6.96  10.95  3.97   5.64
[3,] 990.78 0.72 156.02 829.72 296.23 163.93 258.12 93.47 133.01
```

```
[4,] 341.45 0.25  53.77 285.94 102.09  56.50  88.95 32.21  45.84
```

R-keel lubab sama struktruuriga tabelitega lahter-lahtrilt teheid teha. Leiame tegeliku ja teoreetilise sageduse vahed rühmades

```
> round(ryhmatabel-teoreetiline, 2)
        A    A1     A2      B     B1     B2      C     C1     C2
1   -3.73  0.00  -0.59  -1.12  -1.12  -0.62   6.03   1.65  -0.50
2  -30.05 -0.03  -6.62 -11.21 -10.57  -1.96  19.05  16.03  25.36
3  196.22  0.28  56.98  -1.72  52.77 -76.93 -74.12 -79.47 -74.01
4 -162.45 -0.25 -49.77  14.06 -41.09  79.50  49.05  61.79  49.16
```

Kuna rühmad on eri suurustega, siis jagame arvud veel rühmasuurustega läbi, saab suhtarvu, kuivõrd on vastavat keeletaset rühmas ühtlasest jaotusest rohkem või vähem. Nagu näha, siis algajamate tekstid on koondunud pigem kolmandasse rühma, keskmiste tasemed viimasesse rühma ning edasijõudnute omad teise ja esimesse - viimane neist väike

```
> round((ryhmatabel-teoreetiline)/rowSums(ryhmatabel), 2)
      A A1    A2     B    B1    B2     C    C1    C2
1 -0.34  0 -0.05 -0.10 -0.10 -0.06  0.55  0.15 -0.05
2 -0.24  0 -0.05 -0.09 -0.09 -0.02  0.15  0.13  0.20
3  0.07  0  0.02  0.00  0.02 -0.03 -0.03 -0.03 -0.03
4 -0.16  0 -0.05  0.01 -0.04  0.08  0.05  0.06  0.05
```

Harjutus

- Tehke sama arvutuskäik läbi, jagage andmed nelja asemel kahte rühma.
- Käivitage hii-ruut-test
- Kirjeldate tekstide jaotust keeletasemete järgi rühmadesse
- Naabriga koos jagage andmed kolme rühma ja jälgige tulemust
- Arvestage ainult keeletasemeid A2, B1, B2, C1

Kahe rühma puhul

```
tekstiandmed$ryhm=kmeans(tekstiandmed %>% select(A:Z), centers=2)$cluster
ryhmatabel=tekstiandmed %>% group_by(ryhm, keeletase) %>% summarise(kogus=n()) %>%
spread(keeletase, kogus, fill=0) %>% ungroup() %>% select(A:C2)
teoreetiline=t(sapply(1:nrow(ryhmatabel),
function(ryhmanr){colSums(ryhmatabel)*rowSums(ryhmatabel)[ryhmanr]})/sum(ryhmatabel))
round((ryhmatabel-teoreetiline)/rowSums(ryhmatabel), 2)
```

```
      A A1    A2     B    B1    B2     C    C1    C2
1 -0.22  0 -0.05 -0.08 -0.08  0.07  0.08  0.14  0.14
2  0.03  0  0.01  0.01  0.01 -0.01 -0.01 -0.02 -0.02

> ryhmatabel %>% chisq.test()

        Pearson's Chi-squared test
```

```
data:  .
X-squared = 839.19, df = 8, p-value < 2.2e-16
```

Kolme rühma puhul

```
tekstiandmed$ryhm=kmeans(tekstiandmed %>% select(A:Z), centers=3)$cluster
ryhmatabel=tekstiandmed %>% group_by(ryhm, keeletase) %>% summarise(kogus=n()) %>%
    spread(keeletase, kogus, fill=0) %>% ungroup() %>% select(A2, B1, B2, C1)
teoreetiline=t(sapply(1:nrow(ryhmatabel),
function(ryhmanr){colSums(ryhmatabel)*rowSums(ryhmatabel)[ryhmanr]})/sum(ryhmatabel))
round((ryhmatabel-teoreetiline)/rowSums(ryhmatabel), 2)
ryhmatabel %>% chisq.test()
```

```
> round((ryhmatabel-teoreetiline)/rowSums(ryhmatabel), 2)
     A2    B1    B2    C1
1  0.09  0.11 -0.10 -0.11
2 -0.21 -0.24  0.24  0.20
3 -0.22 -0.34 -0.05  0.61
> ryhmatabel %>% chisq.test()

        Pearson's Chi-squared test

data:  .
X-squared = 474.45, df = 6, p-value < 2.2e-16
```

Nagu näha, siis kolme rühma puhul samuti tulemused üldistatavad, algajad on koondunud esimesse rühma, kesktasemel oskajad teise ning edasijõudnud kolmandasse.

Pythoni statistikakäsklused

R ja Python loetakse siinse materjali kirjutamise ajal andmeteaduse levinumateks keelteks. Kusjuures ettevõetud projekti keele valik sõltub pigem teekidest, mis vajaliku ülesande jaoks vastava keele juures olemas on. Samuti loodava rakenduse juures muudest vajalikest toimingutest. Kui tegemist keeleandmetega, siis eesti keele puhul on üheks kaalukeeleks Pythoni eesti keele paketi estnltk olemasolu - ja annab nõnda põhjuse ka muud tavapärased arvutused selles keeles teha.

T-test

Levinud moodus andmekogumite aritmeetilise keskmise võrdlemiseks. Lihtsaimal juhul ette kaks massiivi:

```
from scipy.stats import ttest_ind
print(ttest_ind([3, 5, 4], [12, 16, 14]))
```

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0503 $ python3.5 ttest1.py
Ttest_indResult(statistic=-7.745966692414834, pvalue=0.0014964810559003347)
```

Väljastatavad kaks väärtust on kogumite aritmeetiliste keskmiste erinevus Studenti hälvete ühikutes (suuremate andmestike puhul lähedane standardhälbele) ning erinevuse olulisust (ehk nullhüpoteesi tõenäosust) näitav P-väärtus

Erinevuste usaldusvahemiku jaoks paar käsklust lisaks

```
import statsmodels.stats.api as sms
cm = sms.CompareMeans(sms.DescrStatsW([3, 5, 4]), sms.DescrStatsW([12, 16, 14]))
print(cm.tconfint_diff(usevar='unequal'))
```

väljundiks

```
(-14.155399503183473, -5.8446004968165273)
```

ehk siis 95% tõenäosusega on sisestatud andmete põhjal üldistades esimese arvukogumiku aritmeetiline keskmine väiksem 14,2 kuni 5,8 ühikut

Harjutus

- Kirjuta ühte stringi mõned sidesõnad ja teise stringi mõned nimisõnad
- Kuva T-testi abil hinnang nende sõnapikkuste keskmise sarnasusele + võimalusel usaldusvahemik

Failist loetud andmed

Tuttav fail, võrdleme Kungla rahva ning lambipirni loo sõnade keskmisi pikkusi

```
from scipy.stats import ttest_ind
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
kunglapikkused=sonad[sonad.lugu=="kungla"].sonapikkus
lambipirnipikkused=sonad[sonad.lugu=="lambipirn"].sonapikkus
print(ttest_ind(kunglapikkused, lambipirnipikkused))
```

Näeb Kungla rahva sõnade pikkusi tulbana (algus ja ots) ning arvuloeteluna. Lõppu kahe loo pikkuste võrdlus - Vana laululoo sõnad on selgelt lühemad.

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0503 $ python3.5 ttest2.py
0     3
1     6
2     6
3     7
4     3


70    4
71    3
72    6
73    4
74    5
Name: sonapikkus, Length: 75, dtype: int64
[3 6 6 7 3 4 5 4 5 4 9 8 4 6 4 5 4 3 5 7 4 3 6 7 5 6 4 2 7 2 6 9 4 6 4 2 4
```

```
 3 2 5 5 4 4 5 6 9 2 6 5 4 2 8 7 4 3 5 6 4 2 6 6 3 4 2 4 5 4 2 9 6 4 3 6 4
 5]
Ttest_indResult(statistic=-3.3098045648541246, pvalue=0.00098363634739143005)
```

ANOVA

T-test võrdleb kahe rühma aritmeetilisi keskmisi, ANOVA puhul võib rühmi olla rohkem. Kõigepealt uuritakse, et kas keskmiste vahel üldse on üldistatavat vahet. Kui jah, siis saab eraldi leida, et milliste rühmade vahel see avaldub.

```
from scipy import stats
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikk
used_haalikud.txt")
print( stats.f_oneway(
     sonad[sonad.lugu=="kungla"].sonapikkus,
        sonad[sonad.lugu=="lambipirn"].sonapikkus,
        sonad[sonad.lugu=="hinnad"].sonapikkus,
   ))

jaagup@praktika1 ~/public_html/2019/kvantdh/0503 $ python3.5 anova1.py
F_onewayResult(statistic=4.7767456205379144, pvalue=0.0086355063608118555)
```

Vastuseks tuli, et sõnade pikkuse sõltumatuse tõenäosus loost on 0,00863 ehk alla ühe protsendi.

Et seos vähemasti 99% tõenäosusega olemas, siis saab uurida, et milliste paaride vahel see avaldub

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_hinnad_pikk
used_haalikud.txt")
print(pairwise_tukeyhsd(sonad.sonapikkus, sonad.lugu))
```

Käivitamisel paistab, et Kungla rahva laulu sõnad erinevad pikkused üldistataval määral, hindade ja lambipirni teksti puhul pole võimalik nullhüpoteesi ümber lükata ehk erinevust üldistatavaks pidada

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0503 $ python3.5 anova2.py
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=================================================
group1   group2  meandiff  lower   upper  reject
-------------------------------------------------
hinnad   kungla  -1.2152  -2.1831 -0.2473  True
hinnad lambipirn -0.0858  -0.6439  0.4724 False
kungla lambipirn  1.1294   0.2322  2.0267  True
-------------------------------------------------
```

Harjutus

- Looge sõne sidesõnadega, nimisõnadega, tegusõnadega
- Näidake ANOVA abil sõnapikkuse sõltuvust sõnaliigist

```
from scipy import stats
sidesonad='ja ning et'
nimisonad='kassid vanaema raamaturiiul'
tegusonad='jookseb kukub ujub'
pikkusedss=[len(sona) for sona in sidesonad.split()]
pikkusedns=[len(sona) for sona in nimisonad.split()]
pikkusedts=[len(sona) for sona in tegusonad.split()]
print(stats.f_oneway(pikkusedns, pikkusedss, pikkusedts))

from statsmodels.stats.multicomp import pairwise_tukeyhsd
sonad=[]
tyybid=[]
sonad+=pikkusedss
for sona in pikkusedss: tyybid.append("sidesona")
sonad+=pikkusedns
for sona in pikkusedns: tyybid.append("nimisona")
sonad+=pikkusedts
for sona in pikkusedts: tyybid.append("tegusona")

print(sonad, tyybid)
print(pairwise_tukeyhsd(sonad, tyybid))
```

väljund:

```
jaagup@praktika1 ~/public_html/2020/kvantdh/0505 $ python3.5 anova2.py
F_onewayResult(statistic=5.1666666666666696, pvalue=0.049571182075495691)
[2, 4, 2, 6, 7, 12, 7, 5, 4] ['sidesona', 'sidesona', 'sidesona', 'nimisona', 'nimisona',
'nimisona', 'tegusona', 'tegusona', 'tegusona']
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=================================================
 group1   group2  meandiff  lower    upper  reject
-------------------------------------------------
nimisona sidesona -5.6667  -11.0723 -0.2611  True
nimisona tegusona   -3.0   -8.4056   2.4056 False
sidesona tegusona  2.6667  -2.7389   8.0723 False
-------------------------------------------------
```

Hii-ruut test

Alustuseks võrdlus kahe rühma ja kahe tunnusega

```
from scipy import stats
#Mõlemas tekstis 30 vähemalt viietähelist sõna ning 70 alla viie tähega sõna
print(stats.chi2_contingency([[30, 70], [30, 70]])[1])

jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $ python3.5 hii1.py
1.0
```

Nullhüpoteesi tõenäosus on 100%, ehk me ei saa üldistada erinevusi tekstide vahel

Juurde võrdlused sajast kahekümne ning kümne erisuguse tekstiga võrrelduna kolmekümnega.

```
from scipy import stats
print(stats.chi2_contingency([[20, 80], [30, 70]])[1])
print(stats.chi2_contingency([[10, 90], [30, 70]])[1])
```

Kahekümne teksti puhul sajast on 14% võimalus, et tulemus tuleb samast üldkogumist, kust miski juhuvalimi puhul saadi 30 erisugust teksti. Kui leitakse ainult kümme sajast, siis erinevus aga selgelt selge.

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $ python3.5 hii2.py
0.141644690295
0.000782938217891
```

Veidi pikem näide - kuivõrd erineb kuni viietäheliste sõnade sagedus Kungla rahva ning lambipirni loos. Nagu ikka, läheb suurem osa koodist andmete ette valmistamisele

```
import pandas as pd
from scipy import stats
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
kunglaallaviie=len(sonad[(sonad.lugu=="kungla") & (sonad.sonapikkus<5)])
kunglavahemaltviis=len(sonad[(sonad.lugu=="kungla") & (sonad.sonapikkus>=5)])
lambipirnallaviie=len(sonad[(sonad.lugu=="lambipirn") & (sonad.sonapikkus<5)])
lambipirnvahemaltviis=len(sonad[(sonad.lugu=="lambipirn") & (sonad.sonapikkus>=5)])
#Test, kas vähemalt viietäheliste sõnade osakaal erineb üldistatavalt
print(stats.chi2_contingency([[kunglaallaviie, kunglavahemaltviis],
  [lambipirnallaviie, lambipirnvahemaltviis]])[1])
```

Vastuseks, et ka ainuüksi jah/ei vastuste kokku lugemise pealt saab väita, alla viietäheliste sõnade sagedus lugudes on üldistatavalt erinev

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $ python3.5 hii3.py
0.00890271667666
```

Korrelatsioon

Sisse loetud pandas dataframest saab tulpadevahelised korrelatsioonid küsida ühe käsuga

```
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
print(sonad.corr())
```

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $ python3.5 korrelatsioon3.py
               sonapikkus  taishaalikuid  sulghaalikuid
sonapikkus       1.000000       0.901692       0.703999
taishaalikuid    0.901692       1.000000       0.561133
sulghaalikuid    0.703999       0.561133       1.000000
```

Üksikute arvude puhul võib need kahe massiivina ette anda numpy-paketi vastavale käsklusele. Vastus tuleb tabelina, kust vaja sobivast lahtrist otsida. Koos liikuvate andmete puhul on korrelatsiooni koefitsient 1.

```
import numpy as np
print(np.corrcoef([[1, 2, 3], [10, 20, 30]]))
print(np.corrcoef([[1, 2, 3], [1, 4, 3]]))
print("Korrelatsiooni koefitsient: ")
print(np.corrcoef([[1, 2, 3], [1, 4, 3]])[0][1])
```

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $  python3.5 korrelatsioon1.py
 [[ 1.  1.]
 [ 1.  1.]]
[[ 1.          0.65465367]
 [ 0.65465367  1.        ]]
Korrelatsiooni koefitsient:
0.654653670708
```

Massiivi asemel võib ette anda ka tabeli vastava tulba

```
import pandas as pd
import numpy as np
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
print(np.corrcoef([sonad.taishaalikuid, sonad.sulghaalikuid]))
print(np.corrcoef([sonad.taishaalikuid, sonad.sonapikkus])[0][1])
```

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0507 $ python3.5 korrelatsioon2.py
[[ 1.          0.56113311]
 [ 0.56113311  1.        ]]
0.90169218002
```

Harjutus

- Pane korrelatsiooni näited tööle
- Leia sisestatud lause igas sõnas a-tähtede ning e-tähtede arv.
- Leia a- ning e-tähtede sageduse korrelatsioon lause sõnades

Peakomponentide analüüs

Näitena sisendiks Kungla rahva loost sõnade pikkus ning täishäälikute arv. Käsklus

```
tulemus=PCA().fit_transform(kunglaarvud)
```

leiab igale sõnale koordinaadid uues teljestikus

```
import pandas as pd
from sklearn.decomposition import PCA
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
kunglaarvud=sonad[sonad.lugu=="kungla"][["sonapikkus", "taishaalikuid"]]
```

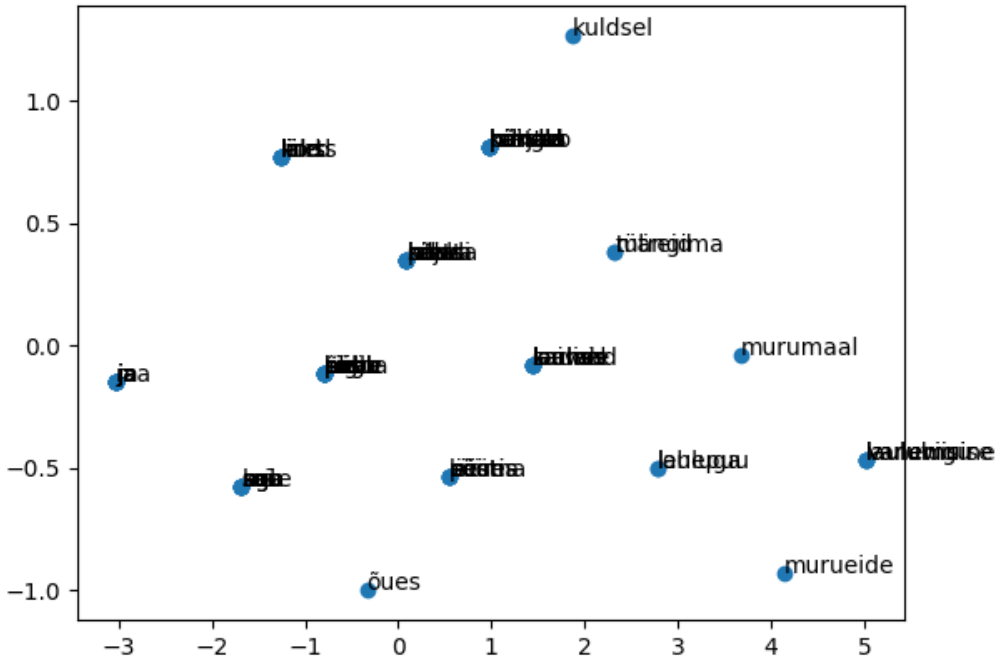
```
print(kunglaarvud)
tulemus=PCA().fit_transform(kunglaarvud)
print(tulemus)
```

Väljund:

```
    sonapikkus  taishaalikuid
0            3              2
1            6              2
2            6              2
3            7              2



[[-1.68430848 -0.57603476]
 [ 0.97681166  0.8090427 ]
 [ 0.97681166  0.8090427 ]
 [ 1.86385171  1.27073518]
--
```

Joonise loomiseks korjame vastustetabelist eraldi välja x-ide ja y-ite massiivi. Nende põhjal joonistame XY-graafiku ning tsükli abil kirjutame sinna juurde vastava sõna teksti.

```
import pandas as pd
from sklearn.decomposition import PCA
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
kunglaarvud=sonad[sonad.lugu=="kungla"][["sonapikkus", "taishaalikuid"]]
tulemus=PCA().fit_transform(kunglaarvud)
xid=[rida[0] for rida in tulemus]
yid=[rida[1] for rida in tulemus]
tekstid=sonad[sonad.lugu=="kungla"].sona.values
plt.scatter(xid, yid)
for nr in range(len(xid)):
  plt.text(xid[nr], yid[nr], tekstid[nr])
plt.savefig("pca2.png")
#plt.show()
```



Harjutus

- Tee näide läbi
- Kuva ainult punktid, ilma sõnadeta
- Kuva joonisele ka lambipirni sõnade punktid, aga teise värviga
- Leia sisestatud lause sõnadest a-, e-, i- ja o-tähtede arv. Arvuta peakomponendid. Koosta XY-joonis kahe esimese komponendi põhjal

Multidimensionaalne skaleerimine

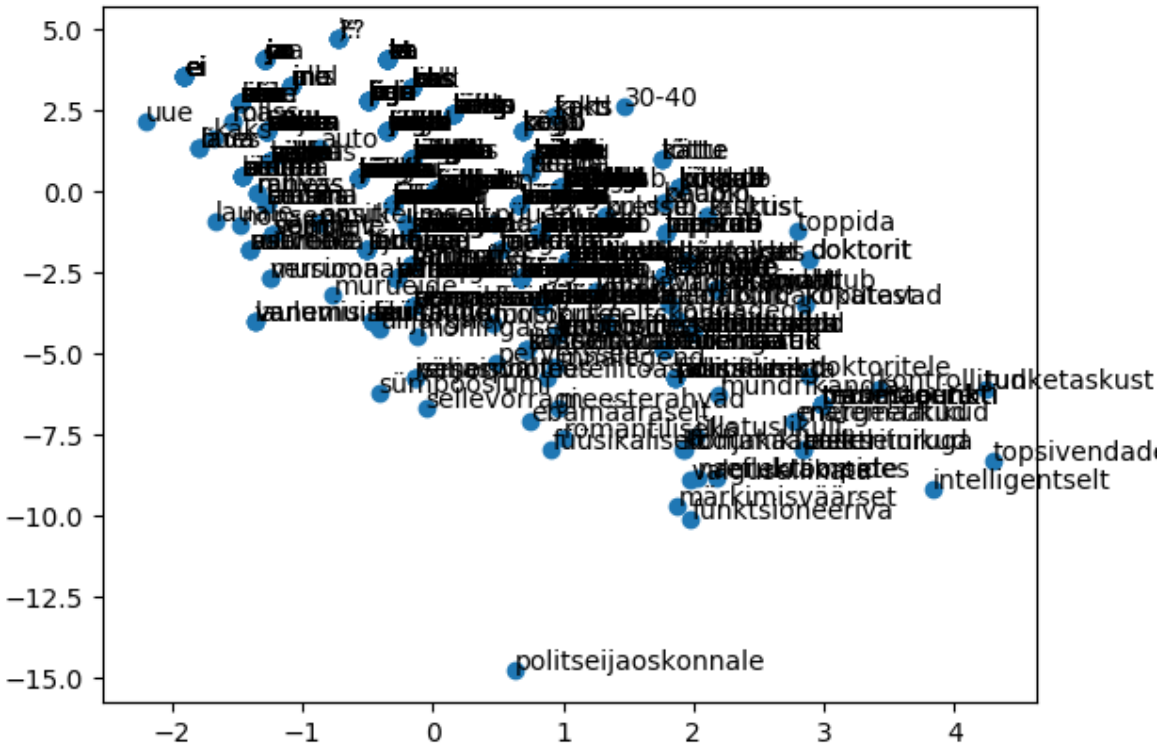
Peakomponentide analüüsiga võrreldes matemaatiliselt vabam meetod, kuid programmeerimiskeeles käivitamine ja tulemuste vaatamine näeb küllalt sarnane välja

```
import pandas as pd
from sklearn.manifold import MDS
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
arvud=sonad._get_numeric_data()
print(arvud.head())
asukohad=MDS().fit_transform(arvud)
print(asukohad)

xid=[rida[0] for rida in asukohad]
yid=[rida[1] for rida in asukohad]
tekstid=sonad.sona.values
```

```
plt.scatter(xid, yid)
for nr in range(len(xid)):
  plt.text(xid[nr], yid[nr], tekstid[nr])
plt.savefig("mds1.png")
```

Väljundis andmed ja joonis

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0510 $ python3.5 mds1.py
   sonapikkus  taishaalikuid  sulghaalikuid
0           3              2              1
1           6              2              2
2           6              2              0
3           7              2              2
4           3              2              0
[[-0.49544325  2.78449471]
 [ 0.98248843  0.11179141]
 [-1.35106894 -0.04862836]
 ...,
 [-1.47057802  2.73803997]
 [-0.31020601 -0.39687389]
 [-0.15701295  0.9797288 ]]
```



Harjutus

- Leidke tekstide sõnadest häälikute a, e, o, u arvud
- Paigutage sõnad MDSi abil joonisele

a-de arv

```
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
sonad["a"]=sonad.apply(lambda rida: rida["sona"].count("a"), axis=1)
print(sonad.head())
```

Väljund

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0510 $ python3.5 mds2.py
     lugu     sona  sonapikkus  taishaalikuid  sulghaalikuid  a
0  kungla      kui           3              2              1  0
1  kungla   kungla           6              2              2  1
2  kungla   rahvas           6              2              0  2
3  kungla  kuldsel           7              2              2  0
4  kungla      aal           3              2              0  2
```

Tulpade lisamine tsükli abil

```
import pandas as pd
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
uuritavad=["a", "e", "o", "u"]
for uuritav in uuritavad:
  sonad[uuritav]=sonad.apply(lambda rida: rida["sona"].count(uuritav), axis=1)
print(sonad.head())
```

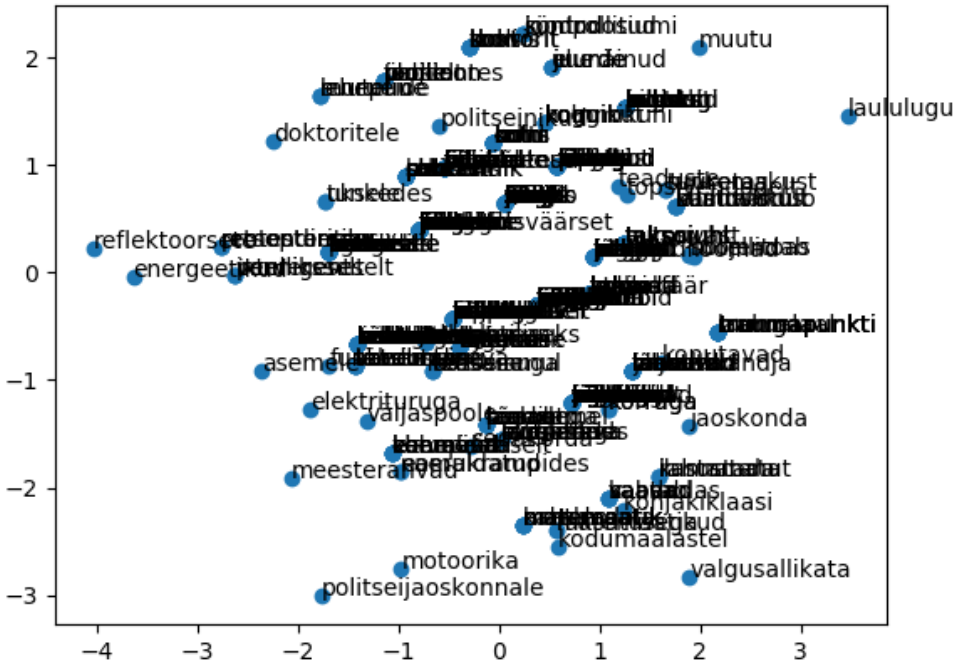
```
jaagup@praktika1 ~/public_html/2019/kvantdh/0510 $ python3.5 mds2.py
     lugu     sona  sonapikkus  taishaalikuid  sulghaalikuid  a  e  o  u
0  kungla      kui           3              2              1  0  0  0  1
1  kungla   kungla           6              2              2  1  0  0  1
2  kungla   rahvas           6              2              0  2  0  0  0
3  kungla  kuldsel           7              2              2  0  1  0  1
4  kungla      aal           3              2              0  2  0  0  0
```

Skaleerimise tulemus joonisena

```
import pandas as pd
from sklearn.manifold import MDS
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
uuritavad=["a", "e", "o", "u"]
```

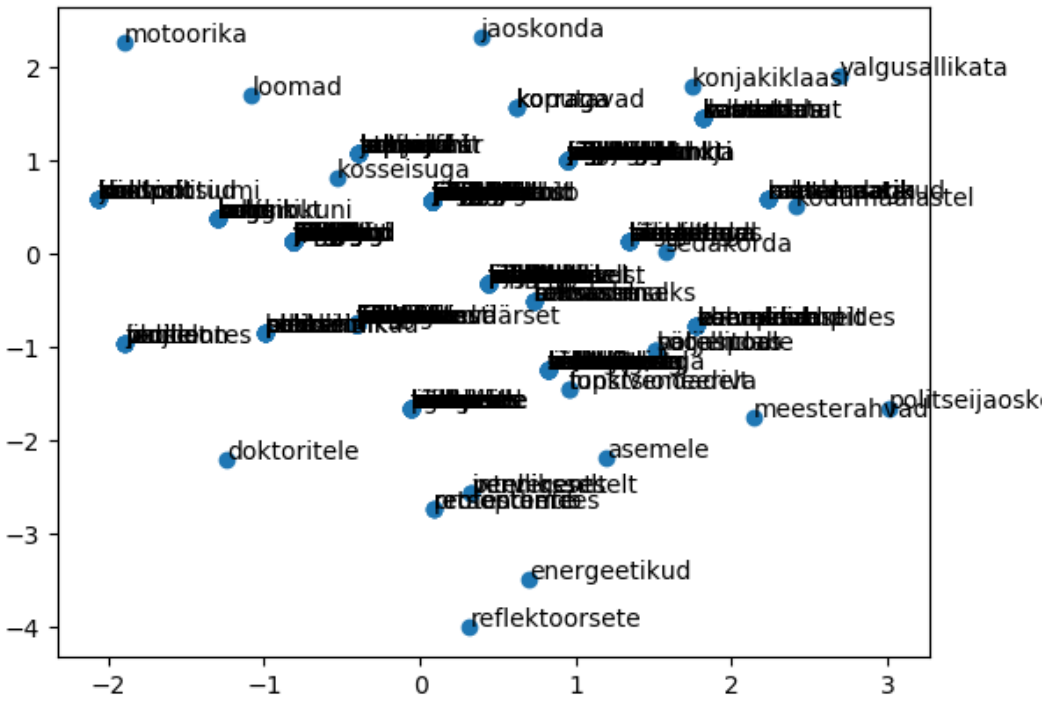
```
for uuritav in uuritavad:
  sonad[uuritav]=sonad.apply(lambda rida: rida["sona"].count(uuritav), axis=1)

asukohad=MDS().fit_transform(sonad[uuritavad])

xid=[rida[0] for rida in asukohad]
yid=[rida[1] for rida in asukohad]
tekstid=sonad.sona.values
plt.scatter(xid, yid)
for nr in range(len(xid)):
  plt.text(xid[nr], yid[nr], tekstid[nr])
plt.savefig("mds2.png")
```

Paremale üles kogunevad siinse joonise puhul pigem u-tähti sisaldavad sõnad, paremale alla a-tähti sisaldavad sõnad



Vahetades uuritavaid saame uue joonise, kus u-tähtede arve ei arvestata

```
uuritavad=["a", "e", "o"]
```



Lineaarne regressioon

Lihtne ja levinud arvulise ennustamise vahend masinõppes. Arvestatakse käsu juures kohe, et sisendis võib olla rohkem kui üks tulp, seetõttu tuleb sisendandmed anda ette eraldi massiividena. Siin näiteks õpetatakse mudelit, et arvudele 20, 30 ja 40 vastavad teisenduse tulemusena arvud 2, 3 ja 4. Vastet küsitakse ennustust sisendandmetele 25 ja 50

```
from sklearn.linear_model import LinearRegression
mudel=LinearRegression().fit([[20], [30], [40]], [2, 3, 4])
print(mudel.predict([[25], [50]]))
print(mudel.coef_)
print(mudel.intercept_)
```

Vastusena pakutakse täiesti usutavalt, et ennustuse tulemused on 2,5 ning 5,0. Leitud valemi kordaja ehk koefitsient on 0,1 ning vabaliige väga lähedane nullile.

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0510 $ python3.5 regressioon1.py
[ 2.5  5. ]
[ 0.1]
4.4408920985e-16
```

Põhjalikumas näites ennustatakse täishäälikute ning sulghäälikute arvu põhjal sõna pikkust. Käsklus suudab vastu võtta ka pandase dataframest tulevad andmed.

```
import pandas as pd
```

```
from sklearn.linear_model import LinearRegression
sonad=pd.read_csv("http://www.tlu.ee/~jaagup/andmed/keel/kunglarahvas_lambipirn_pikkused_ha
alikud.txt")
mudel=LinearRegression().fit(sonad[["taishaalikuid", "sulghaalikuid"]], sonad.sonapikkus)
print(mudel.coef_)
print(mudel.intercept_)
#matemaatika - 6taish, 3sulgh
print(mudel.predict([[6, 3]]))
```

Leiti, et iga täishäälik suurendab sõna pikkust keskeltläbi 1,64 tähe võrra, sulghäälik vähendab 0,75 tähe võrra.
Loodud mudeli põhjal paluti ennustust sõna "matemaatika" pikkusele, kus 6 täis- ning 3 sulghäälikut. Vastuseks tuli täiesti usutav ja sobilik 12,63

```
jaagup@praktika1 ~/public_html/2019/kvantdh/0510 $ python3.5 regressioon2.py
[ 1.64325477  0.75153461]
0.516439810178
[ 12.63057228]
```

Logistiline regressioon

Valitakse kahe võimaluse vahel. Pisinäide valikust suve ja talve vahel vastavalt temeratuurile

```
from sklearn.linear_model import LogisticRegression
temperatuurid=[[15],[20],[-5],[1]] #temperatuurid
aastaajad=[1, 1, 0, 0] #1-suvi, 0-talv

mudel = LogisticRegression()
mudel.fit(temperatuurid, aastaajad)

uuritavadTemperatuurid=[[-15], [-5], [10], [25]]
ennustatavadAastaajad=mudel.predict(uuritavadTemperatuurid)
print(ennustatavadAastaajad)
#[0 0 1 1]
```

Teises näiteks tunnusteks sõna pikkus ja täishäälikute arv. Uuritakse, kas sõna on pigem Kungla rahva või Lambipirni loo moodi. Esimeses pigem lühemad ja suurema täishäälikute osakaaluga.

```
#kui, aal, lauluga, taksojuht, kirurg, raisk
sisend=[[3,2], [3,2],  [7,4],  [9,3],  [6,2],  [5,2]]
lugu=[0, 0, 0, 1, 1, 1] #0-kungla, 1-lambipirn
from sklearn.linear_model import LogisticRegression
mudel=LogisticRegression()
mudel.fit(sisend, lugu)

#suu, kukla, saivad, raksutades
uuritavad=[[3,2], [5,2], [6,3], [10,4]]
ennustatud=mudel.predict(uuritavad)
print(ennustatud)

#[0 1 0 1]
```

Nagu näha, siis "kukla" aimati õigesti Lambipirni alla ning "saivad" Kungla rahva alla - ehkki "saivad" on pikem.

Kui soovitakse teada, kui kindlalt ennustus vastavasse klassi kuulub, siis aitab käsklus predict_proba

```
toenaosustega=mudel.predict_proba(uuritavad)
print(toenaosustega)
```

```
[[ 0.65360676  0.34639324]
 [ 0.39478851  0.60521149]
 [ 0.51171923  0.48828077]
 [ 0.25497739  0.74502261]]
```

Nagu paistab, siis "saivad" oli üsna piiripealne juhtum

Kordamisküsimused

* R-keele võimalused andmetöötluse juures. Käivitusmoodused, levinumad käsud. Muutujad, omistamine. Arvukogumid ja funktsioonid nendega - min, max, mean, median, range, summary, length, head, tail, unique, lugemine failist
* "Vana" R-i joonistuskäsud - hist, plot, text, boxplot
* Pakett tidyverse - vajadus, võimalused, tähtsamad osad. Andmetabeli sisselugemine, sample_n, head, tail, arrange / desc, filter / %in% , mutate, rename, count. Käsk select - tulpade loend, tulpade vahemik, tulba eemaldamine, add_row, add_column. Grupeerimine - vajadus ja võimalused - group_by, mutate grupeerimisel, summarise, summarise_all, summarise_if - ka mitme tulemustulbaga, rename_all, ungroup. Tibble-tüüpi tabeli loomine, dataframe, matrix, vector
* Joonistuspakett ggplot2. Käsu üldkuju, aes, geom_point / jitter, geom_text, geom_boxplot, geom_col / position_dodge, grupeerimine, värvimine vastavalt grupile, geom_line, geom_curve, xlim, ylim, ggtitle. Animatsioonide koostamine
* Andmetabeli pikk ja lai kuju, spread, gather
* Kordused (for), (s)apply
* Testid, prop.test - usaldusvahemik/intervall, usaldusnivoo. Arvupaar, 2X2 tabel, rohkem mõõtmisi
* Hii-ruut test. Kahe tulbaga, rohkemate tulpadega
* T-test, kahe rühma aritmeetiliste keskmiste võrdlemine, tulemuste usaldusintervall. Paarikaupa t-test, ühepoolne t-test.
* ANOVA, rohkem kui kahe rühma keskmiste võrdlemine. Post hoc test.
* Korrelatsioon - kahe arvukogumi vahel, tabeli tulpade vahel, pairs()
* Peakomponentide analüüs, tunnuste kaalud komponentide juures. Dimensioonide vähendamine - valik, kui palju komponente arvestada. Illustreerimine biplot-i abil - tunnused ja objektid ühel joonisel
* Faktoranalüüs
* Mitmemõõtmeline skaleerimine, näited, võrdlus markeritega
* Stilomeetria - kasutatavad meetodid ja vahendid. Väljundjoonised tasandil, läheduste puu
* Regressioon - vabaliige, koefitsient, ruutliige, mudeli loomine, ennustamine.
* Rühmitamine - k-keskmised
* Statilised arvutused Pythoni abil. T-test, ANOVA, Hii-ruut test, korrelatsioon, regressioon, MDS, PCA

Kokkuvõte

Siinsete lehekülgede läbi närimise tulemusena võiksid andmeanalüüsi juures kasutatavad levinumad testid üldjoontes tuttavad olla. Siin piirduti vaid tavapärasemate seadistustega ning enamasti eeldusega, et andmete jaotus sarnaneb normaaljaotusele. R-i abil mängiti moodused veidi pikemalt läbi, Pythoni juures piirduti põhikäskude käivitamisega, millest saab aga ka oma lahendusele sobiva toe moodustada. Enamikke käsitletud meetodeid on aastakümnete jooksul pikemalt uuritud ning nende toimimisest ja erijuhtudest raamatuid kirjutatud. Kui siit saadud arvutuse põhjal põnev järeldus välja paistab, siis pidulikumas kohas selle absoluutse tõena kuulutamiseks on kasulik taustaallikatest üle kontrollida, et kas lähteandmed meetodi pakutavate usalduspiiride kinnitamiseks siiski sobilikud on. Enamasti tuleb kasuks samadele andmetele rakendada mitu meetodit ning vaadata, kuivõrd nende tulemused sarnanevad, milline meetod milliste parameetritega järeldamiseks võimalikult häid vastuseid annab. Seejärel jällegi mõtiskleda, et kuivõrd on leitud seos üldistatav ning kuivõrd rakendatav vaid praegustele andmetele.

Ehkki üldistavad meetodid kõlavad uhkelt, siis tasub enne nende kasutamist ja sõnastamist oma andmetest harilik kirjeldav ülevaade saada - kui palju midagi on, mitu protsenti see moodustab, kuidas on arvuliste andmete jaotus histogrammi järgi. Kui andmetest selgem üldülevaade käes, siis võib paremini mõtiskleda selle üle, et mida mille kaudu järeldada võib. Ning samuti paistavad välja, kui meetodis kasutatud arvutuse tulemus või rakendamine ise nende andmete peal vigane on.

Tänapäevased arvutid suudavad ette valmistatud andmetest vastuse leida sageli sekunditega. Andmete sobivale kujule saamine võib aga tunduvalt rohkem aega võtta. Samuti tasub tõsiselt mõtiskleda, et kuidas käituda puuduvate andmetega. Neist pole enamasti pääsu - olgu siis küsitluste vastuste või mõõdulindiga mõõtmise juures. Kõige lihtsam on vastavad andmeread välja jätta, aga siis tuleb jälgida, et sealjuures "last koos pesuveega" välja ei viska.

Head arvutamist ja tulemuste esitamist!