

Version 2.0
Kalev Pärna 31.05.2024



Statistilise metsainventuuri (SMI) arendamine

Tartu Ülikooli
matemaatika ja statistika instituudi (MSI)
lõpparuanne

TÖÖVÕTULEPING nr 4-1/23/52

Tartu Ülikool

Reg. kood 74001073

Ülikooli 18, Tartu 50090

E-post: info@ut.ee

Versioon 2.0
Kalev Pärna 31.05.2024



Kodulehekül: www.ut.ee

Versiooni nr.	Kuupäev	Muudatus	Autor
1	31.05.2024	Lõpparuanne	Kalev Pärna, TÜ MSI
2	15.07.2024	Lõpparuanne koodi täiendusega	Kalev Pärna, TÜ MSI

Sisukord

1.	Sissejuhatus ja tulemuste kokkuvõte	5
	Tulemuste lühikokkuvõte	5
2.	Maakasutuse muutuste maatriksi koostamine ja kohandamine hetkjaotustega	6
2.1.	Ülevaade probleemist.....	6
2.2.	Probleemi lahenduse skeem	7
2.2.1.	Andmekasutuse skeem.....	7
2.2.2.	Algoritmi kirjeldus (muutuste maatriksi kohandamine etteantud hetkjaotustega)	8
2.3.	Programm ja analüüs.....	9
2.3.1.	Eeltöö - andmete sisselugemine, korrastamine	10
2.3.2.	5-aastaste üleminekute leidmine alalistel traktidel	11
2.3.3.	Ajutiste traktide info kasutamisest.....	14
2.3.4.	1-aastaste üleminekumaatriksite leidmine	18
2.3.5.	Leiame ka libisevate keskmiste abil silutud hinnangud.....	20
2.3.6.	Kooskõla testimine ajalooliste andmetega.....	21
2.3.7.	Kooskõla testimine pikemate perioodide korral	22
2.3.8.	Tabelite vormistamine.....	24
3.	Kasvukohaproovitükkide kasutamine kogutagavara hindamisel.	26
3.1.	Probleemi kirjeldus	26
3.2.	Probleemi lahendusskeem	26
3.3.	Mitmese imputeerimise meetod hektaritagavara hindamiseks	27
3.3.1.	Mitmese imputeerimise meetod.....	27
3.3.2.	Kood.....	28
3.4.	Mitmese imputeerimise rakendus: hektaritagavara hindamine kogu metsamaal	31
3.5.	Mitmese imputeerimise rakendus: hektaritagavara hindamine väikeses osakogumis .	32
Näide 1.	Esimese boniteedi range kaitsega männikud. Keskmise hektaritagavara	32
Näide 2.	Esimese boniteedi range kaitsega kuusikud. Keskmise hektaritagavara.....	33
3.6.	Relaskoobiandmete vajalikkuse analüüs kasvukohaproovitükkidel	34
3.6.1.	Sissejuhatus	34
3.6.2.	Analüüsimeetod ja kood	34
3.6.3.	Hektaritagavara hindamine kogu metsamaal kasutades ka relaskoobiandmeid.....	38
3.6.4.	Hektaritagavara hindamine kogu metsamaal ilma relaskoobiandmeteta	38
3.6.5.	Kokkuvõte (relaskoobimõõtmiste vajalikkus kasvukohtadel)	39



3.7.	Kasvukohaproovitükkide vajalikkuse täiendav testimine tagavara hindamisel	39
3.8.	Kasvukohaproovitükkide täiendav testimine väikeste osakogumite hindamisel	40
3.8.1.	Relaskoobinäidu vajalikkuse testimine imputeerimisel (väike osakogum)	40
3.8.2.	Kasvukohatükkide vajalikkuse testimine tagavara hindamisel (väike osakogum)	41
3.9.	Kokkuvõte	42
4.	Optimaalne ajaaken mudelite hindamiseks	43
4.1.	Sissejuhatus	43
4.2.	Eelanalüüs: h100 keskmise muutumine ajas	43
4.3.	Täiendav metoodiline analüüs H100 muutuste uurimisel	49
4.3.1.	H100 mudeli kontroll	50
4.3.2.	Kõrguse kasv vanuserühmade kaupa	50
4.3.3.	H100 muutused mudelpuudel	51
4.3.4.	Kokkuvõte H100 ja kõrguse dünaamika analüüsist	53
4.4.	Optimaalse ajaakna leidmine kõrguse mudeli jaoks	53
4.4.1.	Sissejuhatus	53
4.4.2.	Erineva pikkusega ajaakende võrdlus kõrgusmudeli korral	53
4.4.3.	Viieaastase ajaakna võrdlus senise mõõteajalooga	55
4.4.4.	Kokkuvõte	57
4.4.5.	Programmi kood (joonised 11 ja 12)	58
4.4.6.	Programmi kood (joonised 13 ja 14)	65
4.5.	Optimaalse ajaakna leidmine mahumudeli jaoks	74
4.5.1.	Analüüsi skeem	74
4.5.2.	Tulemused ja kokkuvõte	75
4.5.3.	Programmi kood	76
5.	Kasutatud kirjanduse loetelu	82
6.	Lisa	83

1. Sissejuhatus ja tulemuste kokkuvõte

Lõpparuanne käsitleb TÜ MSI projektimeeskonna poolt saadud tulemusi SMI arendusprojekti ülesannete¹ täitmisel ajaperioodil 24.07.2023 - 31.05.2024.

Projekti raames oli peaarõhk pandud SMI-s kasutatava statistilise meetodika edasisele täiendamisele, olles mitmes mõttes jätkuks projektile “*Statistilise metsainventuuri (SMI) traktivõrgu analüüs ja arvutusmeetoditega seotud uuring*” (mai 2022 - jaan. 2023) [2].

Projektimeeskonna käsutuses olid SMI seni kogutud andmed ja ametlikud tööjuhendid (SMI arvutusmeetodiline juhend, SMI välitööde juhend). Töö käigus konsulteeriti SMI spetsialistidega Keskkonnaagentuurist. Teadusliku kirjanduse kaudu tutvuti teiste riikide teadlaste tööga projekti puudutavate teemade vallas. Projektimeeskonnal oli võimalus töö käigus kohtuda ja konsulteerida paljude teiste maade metsastatistikutega (NFI Workshop Birmensdorfis (Šveits), 22.-23.04.2024).

Tulemuste lühikokkuvõte

1. **Maakasutuse muutuste maatriksi koostamine.** On koostatud uus meetodika SMI andmetel aastase maakasutusmuutuste maatriksi koostamiseks vastavalt IPCC nõuetele. Uus meetod tagab maakasutuse muutuste info täieliku kooskõla maakasutuse hetkjaotustega enne ja pärast üleminekuperioodi. Meetod kasutab statistikas tuntud IPFP protseduuri (Iterative Proportional Fitting Procedure), mis minimaalsete paranduste abil teisendab esialgse, potentsiaalselt mittesobitava üleminekumaatriksi kujule, mis on täpses kooskõlas etteantud hetkjaotustega. Meetod võimaldab maakasutuse muutuste analüüsi läbi viia ka mullatüüpide lõikes. Põhjalikult on käsitletud ka ajutistelt traktidelt kogutava info kasutusvõimalusi muutuste maatriksi koostamisel ja edasises LULUCF aruandluses.

¹ Projekti *Statistilise metsainventuuri (SMI) arendamine lähteülesanded* (väljavõte):

- 1) Töötada välja uus meetodika SMI andmetel aastase maakasutusmuutuste maatriksi koostamiseks (arvestades mullatüüpe ja kuivenduse olemasolu) vastavalt IPCC nõuetele. Meetodika peaks sisaldama nii SMI proovitükkide maakasutuse muutuse infot kui ka maakasutuse hetkeolukorra infot, s.t maakasutuse muutuse infot tuleks kalibreerida hetkeolukorra infoga.
- 2) Töötada välja uus meetodika SMI kasvukohaproovitükkide kasutamiseks tagavaraliste tunnuste (hektaritagavara ja tagavara) arvutamisel. Analüüsida tagavaraproovitüki tagavara ja rinnaspindala omavahelist sõltuvust.
- 3) SMI andmete ajaline kasutus mudelites (kõrgusköver ja proovitüki hektaritagavara). Analüüsida, kui võrd pikka aegrida tuleks mudelite koostamiseks ja hindamiseks kasutada. Koostada arvutuseeskiri mudelite koostamiseks.

- 2. Kasvukohaproovitükkide kasutamine kogutagavara hindamisel.** Selgitati välja, et kasvukohaproovitükkidelt kogutava info oskuslik kasutamine aitab mõnevõrra (12-15%) täpsustada puistu tagavaraliste tunnuste (hektaritagavara ja tagavara) hinnangut. Selleks otstarbeks on välja töötatud uus meetodika, mis põhineb mitmesel imputeerimisel ehk kasvukohaproovitükkide mahtude paljukordsel prognoosimisel. Meetod kasutab ära suurt hulka kasvukohaproovitükkidel mõõdetavaid tunnuseid, kusjuures selgitati välja, et ei ole vajadust lisada kasvukohtadel mõõdetavate tunnuste hulka Bitterlichi relaskoobi näitu. Uus meetod võimaldab ka nõuetekohast veaarvutust kogutagavara hinnangule (mis oli seni SMI-s kasutatud mahumodelite puudus).
- 3. Optimaalne ajaaken mudelite hindamiseks.** Uuriti, kui pikka aegrida tuleks kõrguse ja hektaritagavara mudelite kalibreerimisel kasutada, et ühelt poolt käia kaasas looduslike tingimuste sh kliima muutustega (mudeli süstemaatilise vea vähendamise eesmärgil), aga teiselt poolt arvestada, et liiga lühike andmeintervall suurendab mudeli vea juhuslikku komponenti. Leiti, et kõrguse ja mahumodelite rek calibreerimiseks on optimaalne kasutada viimase 6 või 5 aasta andmeid, mis tagab mudelitele keskmiselt suurima prognoositäpsuse RMSE mõttes (RMSE võtab kokku nii prognoosi süstemaatilise kui ka juhusliku vea).

2. Maakasutuse muutuste maatriksi koostamine ja kohandamine hetkjaotustega

2.1. Ülevaade probleemist

SMI tulemuste üks olulisi rakendusvaldkondi on sisendi andmine LULUCF (Land Use, Land Use Change and Forestry) sektori kasvuhoonegaaside emissiooni ja sidumise arvestusse². Selleks määratakse SMI välitööde käigus igale proovitükile vastav maakategoria, mis on üks järgmisest: metsamaa (F), põllumajandusmaa (C), rohumaa (G), märgala (W) (sh ka turbakarjäärid), asustusala (S) ja muu maa (O) (aruandluses eristatakse veel ka turbaväljade kategooriat (P)). Alalise proovitüki maakategoriat saab võrrelda sama proovitüki 5 aasta tagase kategooriaga ning nende erinevuse korral saab registreerida maakasutuse muutuse. (Väikest ebatäpsust võivad seejuures tekitada alalised kasvukohaproovitükid, kus GPS süsteemi mõõtemääramatuse tõttu pole tagatud, et nüüd ja 5 aasta tagasi määrati maakategoriad täpselt sama asukoha põhjal). Ajutiste

² 2006 IPCC Guidelines for National Greenhouse Gas Inventories. Vol. 4: Agriculture, Forestry and Other Land Use. Prepared by the National Greenhouse Gas Inventories Programme, Eggleston H.S., Buendia L., Miwa K., Ngara T. and Tanabe K. (eds). Published: IGES, Japan.

proovitükkide korral aga viie aasta tagusega võrdluse võimalus puudub üldse ning maakasutuse muutusi ja muutuste aega hinnatakse kaardiinfo või aerofotode põhjal. Maakasutuse muutuste alamärkimine ja (kohati paratamatu) eksimine muutuste ajaga tekitavad olukorra, kus muutuste info ja maakategoriate hetkjaotused (marginaaljaotused) ei ole omavahel kooskõlas, s.t. aasta i muutuste maatriksi rakendamine sama aasta marginaaljaotusele ei vii meid järgmise aasta $i + 1$ marginaaljaotuseni, nagu seda oleks oodata täpselt registreeritud muutuste korral. Samasugune ebakõla muutuste info ja hetkjaotuste vahel tekib siis, kui liigume ajas tagasi, s.t. tahame muutuste maatriksi abil jõuda aasta i hetkjaotuselt aasta $i - 1$ hetkjaotuseni. Seega on probleem selles, kuidas täpsustada SMI andmetelt hinnatud muutuste maatrikseid nii, et nad oleksid kooskõlas maakasutuse marginaaljaotustega erinevatel aastatel. Kuna maakasutuse marginaaljaotusi võib lugeda suhteliselt täpselt infoks võrreldes muutuste infoga, siis omavahelise kooskõla saavutamiseks on loogiline kohandada just muutuste maatrikseid. Samal ajal on otstarbekas marginaaljaotusi keskmistada üle 5 aasta, et saavutada nende suurem täpsus.

2.2. Probleemi lahenduse skeem

Allpool on välja pakutud maakasutuse muutuste maatriksi esialgse hindamise meetod SMI proovitükkide põhjal ning sellele järgnev kohandamise (täpsustamise) meetod, mis esialgse üleminekumaatriksi võimalikult väikeste muudatuste abil tagab selle täieliku kooskõla teadaolevate marginaaljaotustega ülemineku eelsel ajal ja ülemineku järgsel ajal. Meetod tagab täpse kooskõla alg- ja lõppjaotustega ka pikema ajaperioodi jooksul.

2.2.1. Andmekasutuse skeem

Lihtsuse mõttes esitame oma meetodi juhul, kus maakategoriate muutuste maatriks hinnatakse üksnes alaliste traktide (nii tagavara- kui ka kasvukohaproovitükid) alusel³, võrreldes antud hetke maakategoriat 5 aasta tagusega. Maakategoria muutusi jälgitakse nii jaotamata kui ka jaotatud proovitükkide piires. Ajutisi trakte ei ole esialgu muutuste maatriksi hindamisel kasutatud, sest nagu näitas analüüs, on seal maakategoria muutuste info madalama kvaliteediga (eriti muutuse aja osas, aga probleemiks on ka muutuste alamärkimine). Samas ajutisi proovitükke kasutatakse maakategoriate marginaaljaotuste hindamisel (koos alalistega), kus nad annavad olulist lisatäpsust. Nii muutuste maatriksid kui ka marginaaljaotused on keskmistatud üle 5 aasta, mis tagab hinnangutes suurema stabiilsuse.

³ Alajaotuses 2.3.3 on eraldi käsitletud juhtu, kus muutuste maatriksi kokkupanekul võib kasutada ka ajutistelt proovitükkidelt kogutud muutuste infot.

2.2.2. Algoritmi kirjeldus (muutuste maatriksi kohandamine etteantud hetkjaotustega)

Meetod koosneb neljast etapist:

1) 5-aastase üleminekumaatriksi leidmine:

Alaliste traktide põhjal leitakse maakateooriate muutuste maatriksid. Selleks võrreldakse mõõtmisaasta i proovitükkide kategooriaid 5 aasta taguse kategooriaga ning saadavad maatriksid keskmistatakse üle 5 aasta. Tulemuseks on 5-aastaste muutuste maatriks ehk üleminekumaatriks P_i^5 . Keskmistamine tagab üleminekumaatriksi suurema täpsuse.

2) 1-aastase üleminekumaatriksi leidmine:

Saadud 5-aastaste muutustemaatriksi P_i^5 matemaatilise juurimise teel leitakse 1-aastane üleminekumaatriks P_i aasta i jaoks. (Selgituseks, 1-aastane üleminekumaatriks P_i on selline maatriks, mida viis aastat järjest rakendades saadakse esialgne 5-aastane üleminekumaatriks P_i^5).

3) Marginaaljaotuste leidmine:

Kõigi proovitükkide (alalised ja ajutised, tagavara- ja kasvukoha) baasil leiame maakateooriate jaotused iga üksikul mõõtmisaastal ning keskmistame need üle viimase viie aasta, et vähendada juhuslikku viga. Tulemuseks on aasta i maakateooriate marginaaljaotus (= hetkjaotus) π_i . Lahendamist vajav probleem seisneb selles, et marginaaljaotuste π_i ja π_{i+1} erinevus ei ole reeglina kooskõlas muutustemaatriksiga P_i . Matemaatiliselt kirjapanduna tähendab see, et $\pi_i P_i \neq \pi_{i+1}$. (Ebakõla põhjuseks on asjaolu, et üleminekumaatriksid ja marginaaljaotused on hinnatud erinevate andmete pealt, lisaks on muutused alamärgitud).

4) Üleminekumaatriksite kohandamine marginaaljaotustega:

Kooskõla saavutamiseks modifitseerime üleminekumaatriksit P_i spetsiaalse meetodiga, milleks on valitud IPFP (Iterative Proportional Fitting Procedure). IPFP on üks neljast selleotstarbelise meetodist (vt Little & Wu(1991)), mis tagavad täpse kooskõla maakateooriate etteantud marginaaljaotustega aastatel i ja $i+1$. IPFP meetod seisneb esialgse pindalalise üleminekumaatriksi ridade ja veergude vaheldumisi ümbernormeerimises, kus igal sammul saavutatakse kooskõla ühe (rea või veeru) marginaaljaotusega (Deming & Stephan (1940)). Protseduuri käigus toimub ümbernormeerimine suur arv kordi (sest ridade ümbernormeerimisega halveneb kooskõla veerumarginaalidega ja vastupidi), koondudes lõpuks maatriksiks, mis on

täielikus kooskõlas etteantud marginaaljaotustega. IPFP meetod püüab säilitada esialgse ülemineku maatriksit P_i struktuuri nii palju kui võimalik ja teeb seal marginaaljaotustega kooskõlla viimiseks ainult minimaalseid muudatusi.

Kommentaar 1: Täpne kooskõla pindalajaotustega säilib ka siis, kui muutuste maatrikseid rakendada üksteise järel mitu korda. Näiteks, kui rakendada aasta i pindalajaotusele π_i järjestikku muutuste maatrikseid P_i , P_{i+1} ja P_{i+2} saame tulemuseks aasta $i+3$ pindalajaotuse π_{i+3} . (Raporti lõpus on ka sellekohased näited).

Kommentaar 2: IPFP meetod annab konkureerivate meetoditega (suurima tõepära (MLRS), miinimum hii-ruut (MCSQ) ja vähimruutude meetod (LSQ)) väga lähedased tulemused, mistõttu võib põhimõtteliselt kasutada ükskõik millist nimetatud meetodit (Little & Wu (1991)).

2.3. Programm ja analüüs

Eesmärgiks on luua kooskõlalised muutuste maatriksid (ehk üleminekumaatriksid), mis võtavad arvesse ka mullatüüpe.

Kõigepealt maakateooriate teisendamiseks kasutatavad andmed.

```
library(tidyverse)

## — Attaching packages ————— tidyverse 1.3.2 —
## ✓ ggplot2 3.4.0   ✓ purrr 0.3.5
## ✓ tibble 3.1.8   ✓ dplyr 1.0.10
## ✓ tidyr 1.2.1   ✓ stringr 1.5.0
## ✓ readr 2.1.3   ✓ forcats 1.0.0

## Warning: package 'forcats' was built under R version 4.2.3

## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag() masks stats::lag()

library(mipfp) #iteratiivne kohandamine marginaalidega

## Warning: package 'mipfp' was built under R version 4.2.3

## Loading required package: cmm

## Warning: package 'cmm' was built under R version 4.2.3

## Loading required package: Rsolnp

## Warning: package 'Rsolnp' was built under R version 4.2.3

## Loading required package: numDeriv
##
```

Version 2.0
Kalev Pärna 31.05.2024



```
## Attaching package: 'mipfp'
##
### The following object is masked from 'package:tidyr':
##
## expand

library(sp)# punkti hulknurka kuulumise kindlakstegemiseks

## Warning: package 'sp' was built under R version 4.2.3

require(openxlsx)

## Loading required package: openxlsx

#failis 00_dirparams2_matemaatikutele olevad vastavused
mklcf2 <- c('G', 'W', 'W', 'S', 'S', 'S', 'O', 'O', 'S', 'F', 'F', 'G', 'C', 'C', 'C', 'F', 'P', 'S', 'S')
names(mklcf2) <- c('P', 'S', 'SV', 'A', 'T', 'TR', 'KK', 'Y', 'K', 'MM', 'M', 'RM', 'PR', 'PK', 'PM', 'MV', 'KT', 'PA', 'PP')

lmkNim <- c('F', 'C', 'G', 'W', 'P', 'S', 'O')
muldNim = c("M", "O")
bonNim = c("R", "V")
```

2.3.1. Eeltöö - andmete sisselugemine, korrastamine

Loeme alusandmestikud sisse ja teisendame proovitükiosade andmeid

```
load("../uued_smi_andmed.RData")
prtosa2 <- prtosa|>filter(on_sees == 'J',!is.na(qmaakond),prtliik != 3)
#korrastame/täiendame
prtosa2<- prtosa2|>
left_join(trakt|>select(atrakt_id,tr_tyyp),by="atrakt_id")|>
left_join(prt|>select(aproovitykk_id,prt_id),by="aproovitykk_id")|>
mutate( lulucf_maakat=case_when(
  # Kui c('P','RM','S','KK') puhul on märgitud kas MV või OW, siis korrigeeritakse tüüpi
  maakategooria %in% c('P','RM','S','KK') & frac == "MV" ~ "F",
  maakategooria %in% c('P','RM','S','KK') & frac == "OW" ~ "G",
  is.na(mulla_tyyp)~"W",#tundub, et puuduva mullatüübiga on kõik W alla pandud
  TRUE ~ mklcf2[maakategooria]
),
lulucf_mkendine = mklcf2[maakate_endine],
mulla_tyyp=if_else(is.na(mulla_tyyp),'O',mulla_tyyp) #kui puudu, siis läheb O alla
)
```

Lisame lulucf maakategooriatele mullatüübi info ja jätame välja need vaatlused, kus mullatüüpi ei ole.

```
prtosa2=prtosa2|>#filter(!is.na(mulla_tyyp))|>
mutate(
  lulucf_maakat=paste(lulucf_maakat,mulla_tyyp,sep='_'),
  lulucf_mkendine=ifelse(is.na(lulucf_mkendine),NA,paste(lulucf_mkendine,mulla_tyyp,sep='_'))
)
```

2.3.2. 5-aastaste üleminekute leidmine alalistel traktidel

Leiame esmalt üleminekud neil alalistel traktide proovitükkidel, mida on 5 aastat varem vaadeldud ja mida pole osadeks jagatud. Alalistel traktidel on kaks võimalust: kas arvestada korduskülastuse võrdlemist eelneva külastuse infoga või lähtuda tunnustes *maakat_endine* ja *maakat_muutus* olevast infost. Kuna andmete põhjal tundub, et tunnustes *maakat_endine* ja *maakat_muutus* on muutused tugevalt alamärgitud (vt punkti 2.3.3 lõpus toodud analüüsi pindalamuudatuste kooskõla kohta), on loomulik lähtuda korduskülastuste abil tuvastatavatest muudatustest.

```
lmkMuldNim=sort(paste(rep(lmkNim,each=2),muldNim,sep='_'))
#proovitükid, mida vaadeldi ka 5 aastat tagasi, koos 5 aasta taguste andmetega

muutus1=prtosa2|>
mutate(lulucf_maakat=factor(lulucf_maakat,levels=lmkMuldNim))|>
inner_join(prtosa2|>mutate(aasta=aasta+5),by=c("prto_id","aasta"))|>
filter(grepl(".*0$",prto_id),#ainult jagamata proovitükid, st prto_id lõpus on 0
#prtliik.x==1 #kui kasutada ainult tagavara proovitükke
)|>
group_by(aasta,lulucf_maakat.y,lulucf_maakat.x)|>
summarize(mitu=sum(pindosak.x),groups="drop")|>
rename(aasta_parast=aasta,lcf_mkat_enne=lulucf_maakat.y,lcf_mkat_parast=lulucf_maakat.x)|>
pivot_wider(id_cols=c("aasta_parast","lcf_mkat_enne"),names_from="lcf_mkat_parast",values_from = "mitu",names_or=TRUE,names_expand=TRUE)

aastad_nimekiri=unique(muutus1$aasta_parast)
lcf_mkat_muld_nimed=paste(rep(lmkNim,each=2),muldNim,sep='_')

#garanteerime, et iga aasta ja maakategooria/mullatüübi kombinatsiooni kohta on rida
korrastatud1=expand_grid(aasta_parast=aastad_nimekiri,lcf_mkat_enne=lcf_mkat_muld_nimed)|>
left_join(muutus1,by=c("aasta_parast","lcf_mkat_enne"))|>
arrange(aasta_parast,lcf_mkat_enne)|>
mutate(across(where(is.numeric),~replace_na(.,0)))

korrastatud1|>mutate(across(C_M:W_0,~round(.x,3)))
```

##	#	A	tibble:		266		x		16	
##	aasta_parast	lcf_mkat_enne	C_M	C_O	F_M	F_O	G_M	G_O	O_M	O_O
##		<dbl> <chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
##	1	2004 C_M	328	0	3	0	9	0	1	0
##	2	2004 C_O	0	0	0	0	0	0	0	0
##	3	2004 F_M	0	0	572	4	4	0	0	0
##	4	2004 F_O	0	0	16	147	0	1	0	0
##	5	2004 G_M	31	0	12	1	143	3	1	0
##	6	2004 G_O	1	0	0	1	4	12	0	0
##	7	2004 O_M	1	0	6	0	1	0	23	0
##	8	2004 O_O	0	0	0	0	0	0	0	0
##	9	2004 P_M	0	0	0	0	0	0	0	0
##	10	2004 P_O	0	0	0	0	0	0	0	0
##	#	#	[i] 256		more		rows			
##	#	i 6 more variables:	P_M <dbl>	P_O <dbl>	S_M <dbl>	S_O <dbl>	W_M <dbl>			
##	#	W_0 <dbl>								

Märkus: Tehnilistel põhjustel on viimasest tabelist puudu 6 veergu (kokku peab olema 14=7x2 veergu). Nii see tabel kui ka kõik järgnevad LULUCF teemat puudutavad tabelid on täies mahus loetavad **HTML formaadis**, vt lisatud fail *Lulucf_4.html*

Lisame info selliste üleminekute kohta, kus proovitükk on kas enne või pärast jaotamata.

```
muutus2=prtosa2|>
mutate(lulucf_maakat=factor(lulucf_maakat,levels=lmkMuldNim))|>
inner_join(prtosa2|>mutate(aasta=aasta+5),by=c("prt_id","aasta"),relationship = "many-to-many")|>
filter(osa_nr.x==0 & osa_nr.y>0 | osa_nr.x>0 & osa_nr.y==0#, #ainult juhud, kus enne või pärast on jagamata
#prtliik.x==1 #kui kasutada ainult tagavara proovitükke
)|>
mutate(pindosak=pmin(pindosak.x,pindosak.y))|>
group_by(aasta,lulucf_maakat.y,lulucf_maakat.x)|>
summarize(mitu=sum(pindosak),groups="drop")|>
rename(aasta_parast=aasta,lcf_mkat_enne=lulucf_maakat.y,lcf_mkat_parast=lulucf_maakat.x)|>
pivot_wider(id_cols=c("aasta_parast","lcf_mkat_enne"),names_from="lcf_mkat_parast",values_from = "mitu",names_s
ort=TRUE,names_expand=TRUE)

korrastatud2=expand_grid(aasta_parast=aastad_nimekiri,lcf_mkat_enne=lcf_mkat_muld_nimed)|>
left_join(muutus2,by=c("aasta_parast","lcf_mkat_enne"))|>
arrange(aasta_parast,lcf_mkat_enne)|>
mutate(across(where(is.numeric), ~replace_na(., 0)))

korrastatud2|>mutate(across(C_M:W_O,~round(.,3)))

## # A tibble: 266 × 16
##   aasta_parast lcf_mkat_enne   C_M   C_O   F_M   F_O   G_M   G_O   O_M   O_O
##   <dbl> <chr>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     2004 C_M           2.86    0  0.833  0    0.652  0    0    0
## 2     2004 C_O            0      0  0      0    0      0    0    0
## 3     2004 F_M           0.668    0 17.8    0    0.439  0    0    0
## 4     2004 F_O            0      0  1    1.88  0      0    0    0
## 5     2004 G_M           0.678    0  5.69   0    2.03   0    0    0
## 6     2004 G_O            0      0  0    0.053  0    0.75  0    0
## 7     2004 O_M            0      0  0      0      0      0    0    0
## 8     2004 O_O            0      0  0      0      0      0    0    0
## 9     2004 P_M            0      0  0      0      0      0    0    0
## 10    2004 P_O            0      0  0      0      0      0    0    0
## # [i] 256 more rows
## # [i] 6 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## #   W_O <dbl>
```

Nüüd lisame info kõigilt sellistelt proovitükkidelt, mis olid jaotatud nii praegu kui ka 5 aastat tagasi täpselt samadeks osaproovitükkideks (koordinaatide järgi kontrollides).

```
#abitabel jaotatud_tykid on eraldi arvatud
source("proovitykiosade_koordinaatide_arvestamine.R") #abifunktsioonid pindalade arvutamiseks, samuti leiab andme
stiku jaotatud_tykid
muutus3=prtosa2|>
mutate(lulucf_maakat=factor(lulucf_maakat,levels=lmkMuldNim))|>
inner_join(jaotatud_tykid|>filter(puudu==0),by=c("prt_id","aasta"))|>
left_join(prtosa2|>mutate(aasta=aasta+5),by=c("prto_id","aasta"))|>
```

```
#filter(prtlik.x==1))|>#kui kasutada ainult tagavara proovitükke
mutate(pindosak=pmin(pindosak.x,pindosak.y))|>
group_by(aasta,lulucf_maakat.y,lulucf_maakat.x)|>
summarize(mitu=sum(pindosak),groups="drop")|>
rename(aasta_parast=aasta,lcf_mkat_enne=lulucf_maakat.y,lcf_mkat_parast=lulucf_maakat.x)|>
pivot_wider(id_cols=c("aasta_parast","lcf_mkat_enne"),names_from="lcf_mkat_parast",values_from = "mitu",names_s
ort=TRUE,names_expand=TRUE)

korrastatud3=expand_grid(aasta_parast=aastad_nimekiri,lcf_mkat_enne=lcf_mkat_muld_nimed)|>
left_join(muutus3,by=c("aasta_parast","lcf_mkat_enne"))|>
arrange(aasta_parast,lcf_mkat_enne)|>
mutate(across(where(is.numeric), ~replace_na(, 0)))

korrastatud3|>mutate(across(C_M:W_O,~round(.x,3)))

## # A tibble: 266 × 16
##   aasta_parast lcf_mkat_enne   C_M   C_O   F_M   F_O   G_M   G_O   O_M   O_O
##   <dbl> <chr>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1      2004 C_M           7.5    0  1.12    0    0    0    0    0
## 2      2004 C_O           0      0  0      0    0    0    0    0
## 3      2004 F_M           0      0 62.9   0.796  0    0    0    0
## 4      2004 F_O           0      0  1.16  13.6   0    0    0    0
## 5      2004 G_M           1.30   0  1.4    0   16.8  0.105  0    0
## 6      2004 G_O           0      0  0      0.452  0.156  1.86  0    0
## 7      2004 O_M           0      0  0.221  0      0    0    2.22  0
## 8      2004 O_O           0      0  0      0      0    0    0    0
## 9      2004 P_M           0      0  0      0      0    0    0    0
## 10     2004 P_O           0      0  0      0      0    0    0    0
## # [i] 256 more rows
## # [i] 6 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## # W_O <dbl>
```

Lõpuks lisame info sellistelt proovitükkidelt, kus nii vaatlusaastal kui ka 5 aastat varem oli mitu osa, aga need ei lange vähemalt osaliselt kokku. Selliseid on kõigi vaatluste hulgas küllalt vähe (u. 400).

```
vaatluse_all=jaotatud_tykid|>filter(puudu>0) #jaotatud proovitükid, mille korral jaotus 5 aastat tagasi oli erinev
abitabel=lisa_read(vaatluse_all$prt_id[1],vaatluse_all$aasta[1])#leiame üleminekute pindalad esimese jaotatud proovit
üki jaoks
#tsükkel võtab veidi aega (umbes minut, sõltub arvuti kiirusest)
for(rida in 2:nrow(vaatluse_all)){ #tsükkel üle ülejäänud jaotatud proovitükkide
  abitabel=abitabel|>bind_rows(lisa_read(vaatluse_all$prt_id[rida],vaatluse_all$aasta[rida]))
}

muutus4=abitabel|>
mutate(lcf_mkat_parast=factor(lcf_mkat_parast,levels=lmkMuldNim))|>
group_by(aasta_parast,lcf_mkat_enne,lcf_mkat_parast)|>
summarize(mitu=sum(pindosak),groups="drop")|>
pivot_wider(id_cols=c("aasta_parast","lcf_mkat_enne"),names_from="lcf_mkat_parast",values_from = "mitu",names_s
ort=TRUE,names_expand=TRUE)

korrastatud4=expand_grid(aasta_parast=aastad_nimekiri,lcf_mkat_enne=lcf_mkat_muld_nimed)|>
left_join(muutus4,by=c("aasta_parast","lcf_mkat_enne"))|>
arrange(aasta_parast,lcf_mkat_enne)|>
```

```
mutate(across(where(is.numeric), ~replace_na(., 0)))

korrastatud4|>mutate(across(C_M:W_O,~round(.x,3)))

## # A tibble: 266 × 16
##   aasta_parast lcf_mkat_enne C_M C_O F_M F_O G_M G_O O_M O_O
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2004 C_M 0 0 0 0 0 0 0 0
## 2 2004 C_O 0 0 0 0 0 0 0 0
## 3 2004 F_M 0 0 1.28 0.547 0 0 0 0
## 4 2004 F_O 0 0 0 0.615 0 0 0 0
## 5 2004 G_M 0 0 0 0 0 0 0 0
## 6 2004 G_O 0 0 0 0 0 0 0 0
## 7 2004 O_M 0 0 0 0 0 0 0 0
## 8 2004 O_O 0 0 0 0 0 0 0 0
## 9 2004 P_M 0 0 0 0 0 0 0 0
## 10 2004 P_O 0 0 0 0 0 0 0 0
## # [i] 256 more rows
## # [i] 6 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## # W_O <dbl>
```

Summeerime nüüd kõik alaliste traktide pealt leitud muutuste maatriksid kokku.

```
korrastatud=korrastatud1|>
bind_rows(korrastatud2)|>
bind_rows(korrastatud3)|>
bind_rows(korrastatud4)|>
group_by(aasta_parast,lcf_mkat_enne)|>
summarize(across(C_M:W_O,sum),.groups="drop")
korrastatud|>mutate(across(C_M:W_O,~round(.x,3)))

## # A tibble: 266 × 16
##   aasta_parast lcf_mkat_enne C_M C_O F_M F_O G_M G_O O_M
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2004 C_M 338. 0 4.96 0 9.65 0 1
## 2 2004 C_O 0 0 0 0 0 0 0
## 3 2004 F_M 0.668 0 654. 5.34 4.44 0 0
## 4 2004 F_O 0 0 18.2 163. 0 1 0
## 5 2004 G_M 33.0 0 19.1 1 162. 3.10 1
## 6 2004 G_O 1 0 0 1.50 4.16 14.6 0
## 7 2004 O_M 1 0 6.22 0 1 0 25.2
## 8 2004 O_O 0 0 0 0 0 0 0
## 9 2004 P_M 0 0 0 0 0 0 0
## 10 2004 P_O 0 0 0 0 0 0 0
## # [i] 256 more rows
## # [i] 7 more variables: O_O <dbl>, P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>,
## # W_M <dbl>, W_O <dbl>
```

2.3.3. Ajutiste traktide info kasutamisest

Ajutiste traktide puhul on muudatuste kohta info saadaval ainult tunnustes *maakat_endine*, *maakat_muutus*. Selle info kasutamisel on mitmeid küsimusi:

- 1) Kuidas käsitleda maakat_muutus (ehk muutuse aasta) informatsiooni? Muudatuste tuvastamisel on toimunud ajas muutuseid - perioodil 2009-2015 on vaadeldud ajas tagasi, st muudatus võis olla toimunud rohkem kui 5 aastat tagasi. Selle info korrektseks käsitlemiseks on vaja teada:
 - a) Kas vaadeldavatel aastatel üritati tuvastada kõigi proovitükkide muudatusi ajas (sh ka neil, kus muudatusi pole märgitud) ja kui kaugemale ajas tagasi mindi?
 - b) Kas mindi ajas tagasi kõige viimase muudatuseni või võib eeldada, et vaadeldi ka muudatusele eelnevat ajaperioodi?
 - c) Kas muudatuse aastat võib lugeda täpseks?
 - d) Kas maakategoriate muutumisel võib proovitükke käsitleda sõltumatutena või tuleb arvestada geograafilist kattuvust (erinevate 5-a perioodide ajas tagasi kandmisel võivad tekkida piirkondade ebaühtlane proovitükkidega katmine)?
- 2) Kuidas hinnata võimalikku alamärkimist erinevate muudatuste korral?

Kui ajalist infot kasutada, siis võimalik lähenemine oleks ühest ajutise proovitüki vaatlusest tekitada nõ aastaste vaatluste rida (nt kui vaatlus on 2012, muudatus 2009, eelmine kategooria F, uus G, siis tekitada aastased vaatlused 2008 (F->F), 2009 (F->G), 2010 (G->G), 2011 (G->G) ja 2012 (G->G) jaoks. Sama peaks tegema ka vaatluste korral, kus muutumist pole märgitud (kanda samaks jäämise info selle perioodi peale, mida tagasi vaadati). Kuna aga ajas kaugemale tagasi märgitud muudatuste puhul tekivad siin probleemid stratifitseerimisega, pakume välja lähenemise, kus infot käsitletakse samamoodi kui alaliste traktide proovitükkide korral. Mullatüübi arvestamisel tuleb selleks teha täiendav eeldus, et proovitüki mullatüüp ei muutu.

Seega arvestame ainult viimase 5 aasta peale märgitud muudatusi ja muidu aastat ei arvesta (st kasutame ainult infot, et 5 aasta jooksul on toimunud muutus, ilma märgitud aastat arvestamata).

```
muutus5 = prtosa2|>
  filter(tr_tyyp==2)|> #ajutised traktid
  mutate(
    lcf_mkat_parast=factor(lulucf_maakat,levels=lmkMuldNim),
    lcf_mkat_enne=ifelse(is.na(lulucf_mkendine) | maakat_muutus<aasta-5,lulucf_maakat,lulucf_mkendine)
  )|>
  group_by(aasta,lcf_mkat_enne,lcf_mkat_parast)|>
  summarize(mitu=sum(pindosak),groups="drop")|>
  rename(aasta_parast=aasta)|>
  pivot_wider(id_cols=c("aasta_parast","lcf_mkat_enne"),names_from="lcf_mkat_parast",values_from = "mitu",names_s
ort=TRUE,names_expand=TRUE)

korrastatud5=expand_grid(aasta_parast=aastad_nimekiri,lcf_mkat_enne=lcf_mkat_muld_nimed)|>
  left_join(muutus5,by=c("aasta_parast","lcf_mkat_enne"))|>
```

```

arrange(aasta_parast,lcf_mkat_enne)|>
mutate(across(where(is.numeric), ~replace_na(., 0)))
korrastatud5|>mutate(across(C_M:W_O,~round(.x,3)))

## # A tibble: 266 × 16
##   aasta_parast lcf_mkat_enne   C_M   C_O   F_M   F_O   G_M   G_O   O_M   O_O
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     2004 C_M      527.    0    0    0    0    0    0    0
## 2     2004 C_O        0    0    0    0    0    0    0    0
## 3     2004 F_M      0    0 1047.    0    0    0    0    0
## 4     2004 F_O        0    0    0  204.    0    0    0    0
## 5     2004 G_M        0    0    1    0  141.    0    0    0
## 6     2004 G_O        0    0    0    0    0    6    0    0
## 7     2004 O_M        0    0    0    0    0    0 20.4    0
## 8     2004 O_O        0    0    0    0    0    0    0    0
## 9     2004 P_M        0    0    0    0    0    0    0    0
## 10    2004 P_O        0    0    0    0    0    0    0    0
## # [i] 256 more rows
## # [i] 6 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## # W_O <dbl>

```

Võrdleme viimase 5 aasta summaarset üleminekuinfot, mis on saadud alalistelt traktidelt ja mis on saadud ajutistelt traktidelt.

Alalistelt traktidelt saadud info:

```

keskmistamise_ulus=5
info_alaline=korrastatud|>
group_by(lcf_mkat_enne)|>
mutate(across(C_M:W_O,~stats::filter(.x,filter=rep(1,keskmistamise_ulus),sides=1)))|>
mutate(across(C_M:W_O,~round(.x,3)))|>
drop_na()|>
ungroup()|>
filter(aasta_parast==2022)
info_alaline

```

aasta_parast	lcf_mkat_enne	C_M	C_O	F_M	F_O	G_M	G_O	O_M	O_O
2022	C_M	2601.290	12.000	28.389	0.000	21.522	1.000	2.000	0
2022	C_O	5.000	82.789	0.000	0.044	0.000	2.000	0.000	0
2022	F_M	18.778	0.000	5359.728	67.084	42.578	0.000	8.899	0
2022	F_O	1.000	0.086	50.274	1827.45	1.000	13.612	0.000	0
2022	G_M	55.487	0.000	80.356	0.353	592.560	4.647	1.000	0
2022	G_O	1.000	2.000	1.000	18.062	2.000	150.271	0.000	0
2022	O_M	2.000	0.000	3.361	0.042	4.402	0.000	109.588	0
2022	O_O	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0
2022	P_M	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0
2022	P_O	0.000	0.000	0.000	2.000	0.000	0.000	0.000	0
2022	S_M	12.869	0.626	56.846	3.096	16.603	0.000	3.000	0
2022	S_O	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0
2022	W_M	0.000	0.000	1.284	0.000	0.000	0.000	0.000	0
2022	W_O	3.000	1.000	15.841	16.299	7.079	9.094	0.000	0

Ajutistelt traktidelt saadud info:

```
info_ajutine=korrastatud5|>  
group_by(lcf_mkat_enne)|>  
mutate(across(C_M:W_O,~stats::filter(.x,filter=rep(1,keskmistamise_ulatus),sides=1)))|>  
mutate(across(C_M:W_O,~round(.x,3)))|>  
drop_na()|>  
ungroup()|>  
filter(aasta_parast==2022)  
info_ajutine
```

aasta_parast	lcf_mkat_enne	C_M	C_O	F_M	F_O	G_M	G_O	O_M	O_O
2022	C_M	3189.812	0.00	1.358	0.000	2.000	0.000	0.000	0
2022	C_O	0.000	80.29	0.000	0.000	0.000	0.000	0.000	0
2022	F_M	2.000	0.00	5867.732	0.000	4.000	0.000	2.000	0
2022	F_O	0.000	0.00	0.000	1753.717	0.000	0.000	0.000	0
2022	G_M	9.342	0.00	10.810	0.000	685.868	0.000	0.000	0
2022	G_O	0.000	0.00	0.000	1.864	0.000	128.361	0.000	0
2022	O_M	0.000	0.00	7.085	0.000	0.000	0.000	176.166	0
2022	O_O	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	P_M	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	P_O	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	S_M	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	S_O	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	W_M	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0
2022	W_O	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0

Võrdluse mõttes ka tuvastatud muudatuste arvud:

```
tmp=info_alaline|>select(C_M:W_O)|>as.matrix()  
kokku_alaline=sum(tmp)  
diag(tmp)=0  
muudatusi_alaline=sum(tmp)  
tmp=info_ajutine|>select(C_M:W_O)|>as.matrix()  
kokku_ajutine=sum(tmp)  
diag(tmp)=0  
muudatusi_ajutine=sum(tmp)
```

Alalistel traktidel tuvastati 895.001 (protsentuaalselt 6.86%) proovitükil maakategooria muutust. Ajutistel traktidel tuvastati 73.153 (protsentuaalselt 0.56%) proovitükil maakategooria muutust. Seega näeme, et *ajutistel proovitükkidel on 5-aasta sees toimunud muutusi registreeritud 12 korda vähem kui alalistel proovitükkidel*. Selleks, et vastavate hinnangute kombineerimine oleks mõistlik, tuleks erinevuste põhjusi uurida ja tagada, et muutuste hinnangud oleks vähemalt võrreldavat suurusjärku.

Versioon 2.0
Kalev Pärna 31.05.2024



Kui me siiski otsustame kasutada ajutiste traktide infot, siis üheks võimaluseks on panna kokku ajutiste ja alaliste traktide andmed enne marginaalidega kohandamist (märkida `eval=TRUE` allpool)

```
#märkida eval=FALSE, kui ajutiste proovitükkide infot mitte kasutada
korrastatud=korrastatud|>
  bind_rows(korrastatud5)|>
  group_by(aasta_parast,lcf_mkat_enne)|>
  summarize(across(C_M:W_O,sum),.groups="drop")
```

2.3.4. 1-aastaste üleminekumaatriksite leidmine

Viie aasta pealt hinnatud üleminekud võivad puududa lihtsalt seetõttu, et teatud tüüpi üleminekud on suhteliselt harv sündmus. Nii võib juhtuda, et mõned üleminekud on tegelikult võimalikud, aga hinnatud maatriksis on vastavatel kohtadel nullid. Sellised olukorrad võivad marginaalidega kohandamisel tekitada väikesi lisaprobleeme.

Seetõttu teeme andmeid uurides kindlaks, millised üleminekud tegelikkuses üldse eksisteerivad, ning tekitame maatriksi, mille abil vastavates positsioonides vaatluste põhjal leitud nullid asendatakse mingi väikese arvuga, näiteks 10^{-8} .

```
asendamine=10(-8)
mittenullid=(korrastatud|>group_by(lcf_mkat_enne)|>summarize(across(C_M:W_O,sum))|>select(-lcf_mkat_enne)>0)
*asendamine
```

Silume eeldusel, et 5 aasta jooksul saame terve Eesti üleminekute kohta tervikliku info, st leiame 5-aasta üleminekumaatriksid ainult eelnevatelt andmetelt, kasutades etteantud arvu aastate keskmisi.

```
keskmistamise_ulatus=5

maatriksid_alg=korrastatud|>
  group_by(lcf_mkat_enne)|>
  #summarize_all(funs=mean)|>#sum)|>
  mutate(across(C_M:W_O,~stats::filter(.x,filter=rep(1,keskmistamise_ulatus),sides=1))|>
    drop_na())|>
  ungroup()

reasummad = maatriksid_alg|>select(C_M:W_O)|>as.matrix()|>rowSums()
maatriksid=maatriksid_alg|>
  mutate(across(C_M:W_O,~.x/pmax(reasummad,1)))
#muudame juhul kui pole vaatluseid, tõenäosuse samaks jäämiseks üheks
for(i in which(reasummad==0)){
  maatriksid[i,maatriksid$lcf_mkat_enne[i]]=1
}
```

Definierime funktsioonid viie aasta üleminekute põhjal ühe aasta üleminekute leidmiseks ja marginaalidega normeerimiseks.

```
aastane_maatriks_lihtne=function(M,k=5){  
  #M - maatriks, mille reasummad on 1 (tinglikud tõenäosused üleminekuks 5 aasta jooksul)  
  tmp=M  
  diag(tmp)=0  
  P1=tmp/k  
  diag(P1)=1-rowSums(P1)  
  return(P1)  
}  
aastane_maatriks_juurimine=function(M,k=5){  
  tmp=eigen(M)  
  V=tmp$vectors  
  tulem=V%*%diag(tmp$values^(1/k))%*%solve(V)  
  #asendame ümardamisvigadest tekkida võivad negatiivsed arvud nullidega  
  return(pmax(tulem,0))  
}  
kohanda_marginaalidega=function(M,marginaal_enne,marginaal_praegu,method=meetod){  
  #meetod võib olla lppf või ObtainModelEstimates paketist  
  tmp=method(diag(marginaal_enne)%*%M,target.list=list(1,2),target.data=list(marginaal_enne,marginaal_praegu),it  
er=10000)  
  P1=tmp$x.hat/rowSums(tmp$x.hat)  
  return(P1)  
}
```

Leiame marginaaljaotused.

```
pindosakuid=prtos2|>  
  filter(!is.na(lulucf_maakat))|>  
  group_by(aasta,lulucf_maakat)|>  
  summarize(kokku=sum(pindosak),.groups="drop")|>  
  pivot_wider(id_cols=c("aasta"),values_from="kokku",names_from="lulucf_maakat",names_sort=TRUE,values_fill=0)
```

Leiame jaotuse iga aasta proovitükkide pealt.

```
#leiame marginaaljaotustele vastavad osakaalud. Eelnevalt omistame  
#puuduvatele kategooriatele väikese pindosakute arvu, sest tegelikkuses kõiki kategooriaid esineb  
P0=pindosakuid|>select(C_M:W_0)|>as.matrix()|>pmax(0.01)  
P0=P0/rowSums(P0)  
row.names(P0)=pindosakuid$aasta  
round(P0,3)
```

```
##      C_M  C_O  F_M  F_O  G_M  G_O  O_M  O_O  P_M  P_O  S_M  S_O
## 1999 0.174 0.000 0.389 0.115 0.154 0.010 0.016 0.000 0.000 0.008 0.054 0.000
## 2000 0.180 0.000 0.395 0.132 0.134 0.007 0.015 0.000 0.000 0.013 0.057 0.000
## 2001 0.186 0.000 0.396 0.117 0.130 0.007 0.012 0.000 0.000 0.010 0.066 0.000
## 2002 0.260 0.000 0.385 0.109 0.073 0.007 0.012 0.000 0.000 0.005 0.060 0.000
## 2003 0.246 0.000 0.422 0.131 0.058 0.006 0.007 0.000 0.000 0.003 0.059 0.000
## 2004 0.225 0.000 0.443 0.093 0.080 0.006 0.013 0.000 0.000 0.009 0.066 0.000
## 2005 0.243 0.000 0.415 0.133 0.068 0.006 0.008 0.000 0.000 0.005 0.077 0.000
## 2006 0.233 0.000 0.396 0.145 0.074 0.013 0.010 0.000 0.000 0.007 0.066 0.000
## 2007 0.266 0.000 0.382 0.137 0.061 0.016 0.008 0.000 0.000 0.002 0.070 0.000
## 2008 0.267 0.000 0.414 0.117 0.059 0.008 0.008 0.000 0.000 0.001 0.073 0.000
## 2009 0.241 0.006 0.419 0.129 0.068 0.011 0.012 0.000 0.000 0.001 0.067 0.001
## 2010 0.243 0.004 0.424 0.124 0.054 0.008 0.006 0.000 0.000 0.010 0.074 0.000
## 2011 0.214 0.005 0.426 0.129 0.058 0.014 0.016 0.000 0.000 0.003 0.079 0.000
## 2012 0.240 0.006 0.404 0.127 0.061 0.012 0.010 0.000 0.000 0.006 0.065 0.000
## 2013 0.247 0.005 0.447 0.107 0.054 0.007 0.009 0.000 0.000 0.001 0.072 0.000
## 2014 0.222 0.006 0.463 0.112 0.058 0.011 0.008 0.000 0.000 0.002 0.071 0.000
## 2015 0.246 0.004 0.445 0.118 0.052 0.007 0.008 0.000 0.000 0.005 0.081 0.000
## 2016 0.232 0.003 0.430 0.125 0.049 0.012 0.008 0.000 0.000 0.004 0.087 0.000
## 2017 0.215 0.010 0.410 0.147 0.053 0.013 0.009 0.001 0.000 0.003 0.080 0.000
## 2018 0.205 0.011 0.420 0.145 0.050 0.013 0.010 0.000 0.007 0.001 0.079 0.000
## 2019 0.203 0.005 0.452 0.127 0.056 0.014 0.011 0.000 0.000 0.001 0.081 0.000
## 2020 0.246 0.004 0.427 0.129 0.047 0.010 0.009 0.000 0.000 0.003 0.084 0.001
## 2021 0.218 0.006 0.416 0.143 0.051 0.011 0.010 0.000 0.000 0.000 0.073 0.001
## 2022 0.221 0.009 0.415 0.142 0.053 0.011 0.016 0.000 0.000 0.001 0.071 0.000
##      W_M  W_O
## 1999 0.000 0.079
## 2000 0.000 0.067
## 2001 0.000 0.076
## 2002 0.000 0.089
## 2003 0.000 0.068
## 2004 0.000 0.065
## 2005 0.000 0.046
## 2006 0.000 0.054
## 2007 0.000 0.060
## 2008 0.000 0.052
## 2009 0.000 0.045
## 2010 0.000 0.052
## 2011 0.000 0.057
## 2012 0.000 0.068
## 2013 0.000 0.051
## 2014 0.000 0.048
## 2015 0.000 0.034
## 2016 0.000 0.050
## 2017 0.002 0.057
## 2018 0.030 0.029
## 2019 0.001 0.049
## 2020 0.002 0.040
## 2021 0.000 0.069
## 2022 0.001 0.059
```

2.3.5. Leiame ka libisevate keskmiste abil silutud hinnangud

```
P0_silutud=pindosakuid|>select(C_M:W_O)|>
mutate(across(everything(),~stats::filter(x,filter=rep(1/5,5),sides=1)))|>
drop_na()|>
as.matrix()|>
pmax(0.01) # Asendame nullid marginaalides väikese arvuga
#Algusega tegelemiseks mitu võimalust
#Iisame keskmise üle olemasolevate, st esimene jääb samaks, teine on kahe esimese keskmine jne kuni neljandani.
P0_algus=pindosakuid|>
```

```
slice_min(aasta,n=4)|>  
select(C_M:W_O)|>  
mutate(across(everything(),cummean))|>  
as.matrix()|>  
pmax(0.01) #  
P0_silutud=rbind(P0_algus,P0_silutud)  
P0_silutud=P0_silutud/rowSums(P0_silutud)  
row.names(P0_silutud)=pindosakuid$aasta
```

Tekitame nüüd andmestiku, kus iga aasta jaoks on kohandatud 1-aastased üleminekumaatriksid (mida eelmise aasta marginaalidele rakendades saame täpselt järgmise aasta marginaalid).

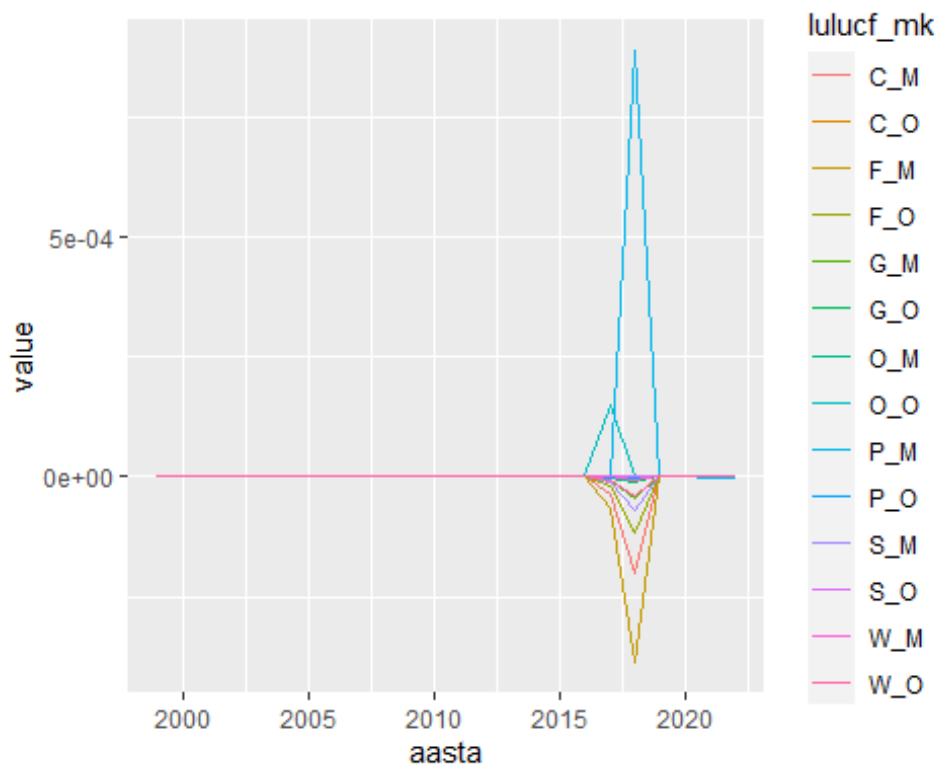
```
meetod = Ipp #ObtainModelEstimates # Ipp # Üleminekumaatriksite kohandamise meetod  
aastane_maatriks=aastane_maatriks_lihtne #katsetame lihtsat versiooni  
esimene_silutud_aasta=min(maatriksid$aasta_parast)  
lisada_aastaid=esimene_silutud_aasta-1999  
maatriksid_aasta=maatriksid|>filter(aasta_parast<esimene_silutud_aasta+lisada_aastaid)|>mutate(aasta_parast=aast  
a_parast-lisada_aastaid)|>bind_rows(maatriksid)  
for(aasta in as.numeric(row.names(P0_silutud))){  
  #print(aasta)  
  marginaal_enne=P0_silutud[as.character(max(aasta-1,1999)),]  
  marginaal_praegu=P0_silutud[as.character(aasta),]  
  P5=maatriksid|>filter(aasta_parast==max(aasta,esimene_silutud_aasta))|>select(C_M:W_O)|>as.matrix()|>pmax(mit  
tenullid)  
  P=kohanda_marginaalidega(aastane_maatriks(P5),marginaal_enne,marginaal_praegu)  
  colnames(P)=colnames(P5)  
  tmp=as_tibble(P)|>add_column(aasta_parast=aasta,lcf_mkat_enne=colnames(P))  
  maatriksid_aasta=rows_update(maatriksid_aasta,tmp,by=c("aasta_parast", "lcf_mkat_enne"))  
}
```

2.3.6. Kooskõla testimine ajalooliste andmetega

Vaatleme, kuidas hinnatud aastased üleminekumaatriksid on ajalooliste andmetega kooskõlas (alates 1999 kuni 2022).

```
korruta_maatriksid_aasta=function(marginaalid,maatriksid_aasta_n,n=1){  
  tulem=as.data.frame(filter(maatriksid_aasta_n,lcf_mkat_enne=="W_M")|>select(-lcf_mkat_enne)) #õige struktuur tule  
muse jaoks  
  for(i in 1:nrow(tulem)){  
    aasta=tulem$aasta_parast[i]  
    P1=maatriksid_aasta_n|>filter(aasta_parast==aasta)|>select(C_M:W_O)|>as.matrix()  
    marginaal_enne=marginaalid[as.character(pmax(aasta-n,1999)),]  
    tulem[i,]=c(aasta,marginaal_enne%*%P1)  
  }  
  return(tulem)  
}  
marginaalid=P0_silutud  
marginaalide_aastad=as.numeric(row.names(marginaalid))  
yleminekud=korruta_maatriksid_aasta(marginaalid,maatriksid_aasta)#marginaalid%*%P  
vahed=marginaalid[marginaalide_aastad %in% yleminekud$aasta_parast,]-yleminekud[,-1]
```

```
ggplot(as_tibble(vahed)|>mutate(aasta=yleminekud$aasta_parast))|>  
  pivot_longer(C_M:W_O,names_to="lulucf_mk"),aes(x=aasta,y=value,color=lulucf_mk))+ geom_line()
```



Joonis 1

See graafik (mille vertikaalteljel on ühikuks tõenäosus) näitab, et 1-aastaste üleminekumaatriksite kooskõla marginaalajaotustega on üldiselt ideaalne, pisike ebakõla on märgata ainult aastatel 2017 ja 2018 (põhjuseks vastuoluline lähteinfo kategooria P kohta).

2.3.7. Kooskõla testimine pikemate perioodide korral

Esitame nüüd paar näidet **pikemate perioodide** üleminekumaatriksite kohta. Konkreetselt vaatleme aastate 2008-2013 ja 2017-2022 üleminekumaatrikseid. Mõlemal juhul 5-aastane üleminekumaatriks leitakse kui viie vastava 1-aastase üleminekumaatriksi korrutis.

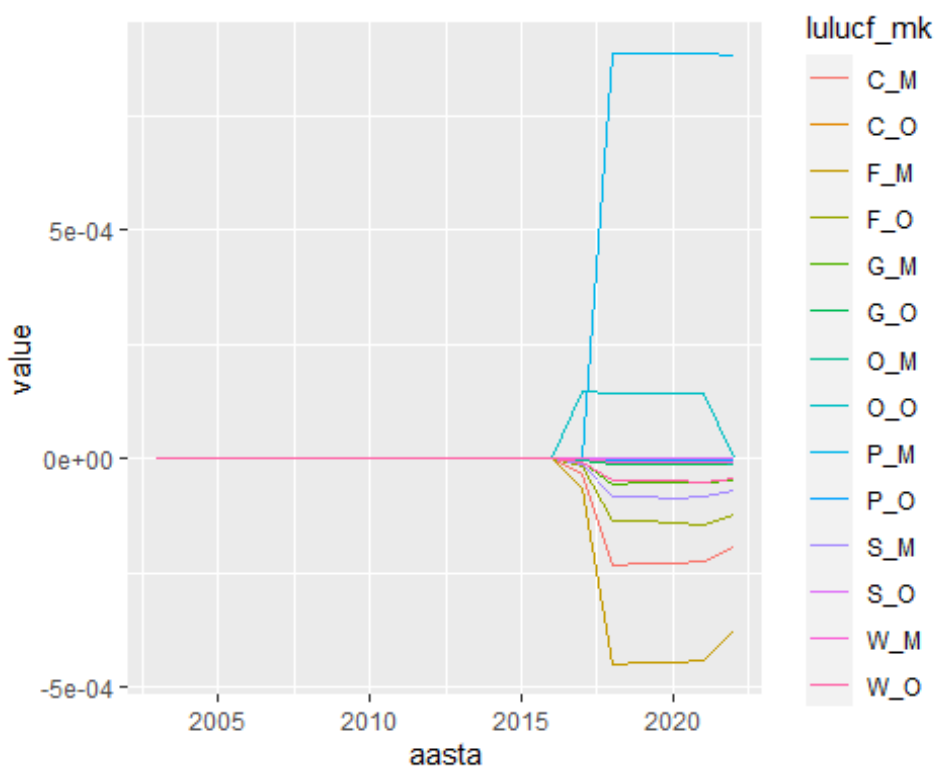
```
Leia_yleminekud=function(maatrksid_aasta,maatrksid_aasta_n,n){  
  tulem=maatrksid_aasta_n|>slice(-(1:(2*length(lmkNim))))  
  aastad=unique(tulem$aasta_parast)  
  for(aasta in aastad){  
    P_n=tulem|>filter(aasta_parast==aasta)|>select(C_M:W_O)|>as.matrix()  
    P_1=maatrksid_aasta|>filter(aasta_parast==aasta-n)|>select(C_M:W_O)|>as.matrix()  
    P=P_1%*%P_n  
    colnames(P)=colnames(P_n)  
    tmp=as_tibble(P)|>add_column(aasta_parast=aasta,lcf_mkat_enne=colnames(P))  
    tulem=rows_update(tulem,tmp,by=c("aasta_parast","lcf_mkat_enne"))  
  }  
}
```

```
}  
return(tulem)  
}
```

Leiame üleminekumaatriksid

```
maatriksid_aasta_2=Leia_yleminekud(maatriksid_aasta,maatriksid_aasta,1)  
maatriksid_aasta_3=Leia_yleminekud(maatriksid_aasta,maatriksid_aasta_2,2)  
maatriksid_aasta_4=Leia_yleminekud(maatriksid_aasta,maatriksid_aasta_3,3)  
maatriksid_aasta_5=Leia_yleminekud(maatriksid_aasta,maatriksid_aasta_4,4)
```

```
yleminekud_5=korruta_maatriksid_aasta(marginaalid,maatriksid_aasta_5,5)#marginaalid%*%P  
vahed=marginaalid[marginaalide_aastad %in% yleminekud_5$aasta_parast,]-yleminekud_5[,-1]  
ggplot(as_tibble(vahed)>mutate(aasta=yleminekud_5$aasta_parast))>  
  pivot_longer(C_M:W_O,names_to="lulucf_mk"),aes(x=aasta,y=value,color=lulucf_mk))+ geom_line()
```



Joonis 2

Graafikult näeme, et perioodidele 2008-2013 ja 2017-2022 vastavate 5-aastaste üleminekumaatriksite kooskõla perioodi algus- ja lõpuaasta marginaalidega on ideaalne, üksnes 2017 tekib vaevumärgatav ebakõla.

```
maatriksid_aasta_5|>filter(aasta_parast==2013)|>mutate(aasta_enne=aasta_parast-5)|>mutate(across(C_M:W_O,~round(.x,3)))
```

```
## # A tibble: 14 x 17
##   aasta_parast lcf_mkat_enne C_M C_O F_M F_O G_M G_O O_M O_O
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2013 C_M 0.933 0.017 0.016 0 0.02 0.001 0.002 0
## 2 2013 C_O 0 1 0 0 0 0 0 0
## 3 2013 F_M 0.003 0 0.979 0.005 0.003 0 0.002 0
## 4 2013 F_O 0 0 0.055 0.915 0 0.002 0 0
## 5 2013 G_M 0.065 0.006 0.115 0.001 0.756 0.01 0.013 0
## 6 2013 G_O 0.011 0.032 0.026 0.051 0.011 0.792 0 0
## 7 2013 O_M 0.002 0 0.054 0 0.028 0 0.88 0
## 8 2013 O_O 0 0 0 0 0 0 0 1
## 9 2013 P_M 0 0 0 0 0 0 0 0
## 10 2013 P_O 0 0 0.019 0.328 0 0.001 0 0
## 11 2013 S_M 0.011 0 0.037 0.004 0.005 0 0.002 0
## 12 2013 S_O 0 0 0.003 0.091 0 0 0 0
## 13 2013 W_M 0 0 0.006 0 0.002 0 0.538 0
## 14 2013 W_O 0.002 0.002 0.014 0.076 0.004 0.026 0.001 0
## # [i] 7 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## # W_O <dbl>, aasta_enne <dbl>
```

```
maatriksid_aasta_5|>filter(aasta_parast==2022)|>
mutate(aasta_enne=aasta_parast-5)|>
mutate(across(C_M:W_O,~round(.x,3)))
```

```
## # A tibble: 14 x 17
##   aasta_parast lcf_mkat_enne C_M C_O F_M F_O G_M G_O O_M O_O
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2022 C_M 0.932 0.008 0.019 0.001 0.011 0.002 0.004 0
## 2 2022 C_O 0.028 0.925 0.001 0.029 0.008 0.008 0 0
## 3 2022 F_M 0.001 0 0.947 0.027 0.004 0 0.004 0
## 4 2022 F_O 0 0 0.01 0.981 0 0.003 0 0
## 5 2022 G_M 0.026 0 0.049 0.005 0.869 0.021 0.002 0
## 6 2022 G_O 0.004 0.004 0.003 0.047 0.007 0.924 0 0
## 7 2022 O_M 0.001 0 0.006 0.008 0.003 0 0.965 0
## 8 2022 O_O 0 0 0 0 0 0 0.987 0.012
## 9 2022 P_M 0 0 0 0 0 0 0 0
## 10 2022 P_O 0.001 0 0.003 0.188 0.001 0 0 0
## 11 2022 S_M 0.006 0 0.024 0.021 0.008 0 0.006 0
## 12 2022 S_O 0 0 0.001 0.111 0 0 0 0
## 13 2022 W_M 0 0 0 0 0 0 0 0
## 14 2022 W_O 0.002 0 0.009 0.038 0.005 0.011 0.001 0
## # [i] 7 more variables: P_M <dbl>, P_O <dbl>, S_M <dbl>, S_O <dbl>, W_M <dbl>,
## # W_O <dbl>, aasta_enne <dbl>
```

2.3.8. Tabelite vormistamine

Lõpuks esitame koodi, mis tekitab tellijale **sobivas formaadis tabelid** aastastest muudatustest. Selle koodi abil on koostatud ka aruandele lisatud tabelid (kokku 14 .xlsx faili ehk 2 faili iga maakategooria kohta, milles üks põhineb ainult alaliste proovitükkide infol, teine aga kasutab nii alaliste kui ka ajutiste proovitükkide infot).

```
eepind <- 4533.9926824946
mkpind=4346.881
```



```
for(lmk in lmkNim){
  lmklmk=paste0(lmk,lmk)
  tolge=c('M'='_min','O'='_org')
  lmkNim_ilma_lmk=lmkNim[lmkNim!=lmk]
  veerud=c('aasta',lmklmk,paste0(lmklmk,tolge),paste0(lmkNim_ilma_lmk,lmk,'_min'),paste0(lmkNim_ilma_lmk,lmk,'_org'))
  M=maatriksid_aasta|>
  mutate(lcfmk=substr(lcf_mkat_enne,1,1))|>
  rename(aasta=aasta_parast)|>
  group_by(aasta)|>
  mutate(across(starts_with(lmk),~.x*P0_silutud[as.character(max(aasta[1]-1,1999))]*mkpind))|>
  ungroup()|>
  select(aasta,lcfmk,starts_with(lmk))|>
  group_by(aasta,lcfmk)|>
  summarize(across(starts_with(lmk),sum),.groups="drop")|>
  pivot_longer(names_to='valja',starts_with(lmk),names_prefix='*_*')|>
  mutate(valjund_nimed=paste0(lcfmk,lmk,tolge[valja]),value=round(value,3))|>
  select(aasta,valjund_nimed,value)|>
  pivot_wider(names_from=valjund_nimed,values_from=value)|>
  mutate({{lmlmk}}:=rowSums(across(starts_with(lmk))))|>
  select(all_of(veerud))
  tblnimi <- paste0("lcfmk_templ2.xlsx")
  tblnimi1 <- paste0("lcfmk_", lmk, "_ajutistega.xlsx")
  wb <- loadWorkbook(tblnimi)
  writeData(wb, sheet = "aastane", M, colNames = T, rowNames = FALSE, startRow = 2)
  saveWorkbook(wb, tblnimi1, overwrite = T)
}
```

Täiendav tehniline info

Aruandele on lisatud järgmised LULUCF teemat puudutavad failid (vt ka Lisa):

- 1) kasutusvalmis LULUCF muutuste tabelid .xlsx formaadis (14 tabelit),
- 2) lugemist hõlbustav fail **Lulucf_4.html** (kvaliteetsem tabelite vormistus),
- 3) täiendav R skript **proovitykiosade_koordinaatide_arvestamine.R** (sisaldab koodi osa, mida aruandes pole toodud).

3. Kasvukohaproovitükkide kasutamine kogutagavara hindamisel.

3.1. Probleemi kirjeldus

Teatavasti kasvukohaproovitükkidel puud ei klupita, mistõttu ei saa otseselt (üksikpuude mahtude liitmise teel) arvutada ka seal paikneva puistu mahtu. Samas kasvukohad annavad olulist infot metsamaa pindala kohta ja seal tehakse ka rida teisi mõõtmisi, mistõttu ei ole mõttekas neid automaatselt välja jätta kogutagavara hindamisest. Ka SMI senine praktika kasutab kasvukohaproovitükke tagavara hindamisel; nimelt koostatakse proovitüki mahumudel $M(H100, A)$, mida rakendatakse nii kasvukoha- kui ka tagavaraproovitükkidel ning selle tulemusena saadakse kogu SMI võrgustikku kattev prognoositud mahuandmestik. Selle meetodi üks oluline puudus on asjaolu, et ta ei võimalda adekvaatset veaarvutust kogutagavara hinnangu jaoks (tekitab vea tugeva alahinnangu, sest mudeliga prognoositud mahtude hajuvus on oluliselt väiksem tegelikust varieeruvusest). Antud projektis on seatud ülesandeks leida võimalusi kasvukohaproovitükkide efektiivsemaks kasutamiseks kogutagavara hindamise eesmärgil. Selleks on välja pakutud mitmese imputeerimise meetod, mida on kirjeldatud allpool.

Teine, eelmisega seonduv probleem seoses kasvukohaproovitükkidega on küsimus, et kas oleks edaspidi vaja kasvukohtadel teha ka relaskoopmõõtmine, mida siis võimalusel kasutada mahumudeli parandamiseks lisaregressori sissetoomise teel. Küsimus on motiveeritud asjaolust, et praegune mahumudel $M(H100, A)$ ei kasuta sisendina ühtki tunnust, mis kirjeldaks puistu tihedust antud proovitükil.

3.2. Probleemi lahenduskeem

Kasvukohaproovitükkide kasutamiseks kogutagavara hindamisel on uuringu tulemusena välja pakutud mitmese imputeerimise meetod – prognoosimeetod, mis arvutab samale proovitükile üksteise järel välja terve seeria erinevaid prognoose (reeglina vähemalt 5 prognoosi, meie uurimuses 20 prognoosi), mistõttu prognoositud andmetesse tekib ka vajalik juhuslikkus ning see võimaldab veaarvutust läbi viia nõutud viisil (Zhang (2003)). Mitmese imputeerimise korral on vaja otsustada, millist algoritmi kasutada üksikprognoosi tekitamiseks. Antud projekti käigus kasutati sel eesmärgil juhumetsa meetodit (Random Forests), mis on piisavalt paindlik suutmaks kasutada suurt hulka argumenttunnuseid, arvestades nii nende omavahelisi seoseid kui ka võimalikke mittelinearseid seoseid sihttunnusega ehk puistu tagavaraga (Breiman (2001)).

Mitmese imputeerimise rakendamisel

- 1) tekitatakse 20 täielikku mahusandmestikku, kus tagavaraproovitükkidel on nende esialgsed arvutatud mahud *arv_maht_es*, kasvukoha-proovitükkidel aga juhumetsa meetodiga prognoositud mahud.
- 2) Iga sellise täisandmestiku põhjal leitakse hektaritagavara punkthinnang ning selle statistiline viga (standardhälve).
- 3) Kõigi punkthinnangute ja nende standardhälvete põhjal leitakse lõpuks mitmese imputeerimise meetodi koondviga.

Mitmese imputeerimise detailsem kirjeldus on toodud allpool.

3.3. Mitmese imputeerimise meetod hektaritagavara hindamiseks

Alljärgnevas esitatakse hektaritagavara hindamise uus skeem, mis seisneb puudevate mahtude mitmeses imputeerimises (prognoosimises) kasvukoha proovitükkidele.

3.3.1. Mitmese imputeerimise meetod

Erinevalt mudelipõhisest lünkade täitmisest konkreetsete väärtustega, mida on siiani kasutatud, on alljärgneva eesmärgiks täita lüngad nii, et 1) eeldused mudeli kujule oleks võimalikult nõrgad ning et 2) leitavad veahinnangud arvestaks võimalikult korrektselt seda, et mudeliga prognoositud väärtused ei ole samaväärsed mõõdetud väärtustega, st tekib täiendav viga seoses prognooside erinevusega tegelikest väärtustest.

Lähtuvalt eelpool kirjeldatud eesmärkidest kasutame mahu prognoosimiseks **mitmese imputeerimise metoodikat**, kusjuures puudevate väärtuste asendamiseks kasutame juhumetsa meetodit (Random Forests). Juhumetsa meetodi kasutamine imputeerimiseks on motiveeritud sellega, et see kohandub automaatselt lokaalsete eripäradega: iga konkreetse vaatluse korral valitakse prognoosimiseks teadaolevate väärtustega vaatluste (s.t. tagavaraproovitükkide) hulgast potentsiaalselt erinevate otsustuskriteeriumite abil (nt mõnes piirkonnas võib olla olulisem geograafiline lähedus, teises aga hoopis mingite muude taustatunnuste komplekti sarnasus). Prognooside juhuslikkust võetakse hindamisel arvesse nii, et lünki täidetakse mitu korda (tihti $m=5$, aga võib kasutada ka suuremat korduste arvu, näiteks meie uurimuses $m=20$) ning saadud hinnangute varieeruvuse ja täieliku andmestiku jaoks hinnatud prognoosi standardhälbe abil selgub saadud koguhinnangu täpsus.

Iga üksiku täieliku andmestiku (tagavaraproovitükkide arvutatud mahud + kasvukohaproovitükkide imputeeritud mahud) põhjal arvutatud mahuhinnangu standardhälbe leidmiseks kasutame eelmises projektis ("Statistilise metsainventuuri (SMI) traktivõrgu analüüs ja arvutus-meetoditega seotud uuring", 2023) pakutud uut metoodikat, mis arvestab ka ruumiliste korrelatsioonidega (nn 'soome meetod').

Mitmese imputeerimise meetodi koondviga (*viga_imp*) koosneb kahest komponendist, millest üks on 20 üksikandmestiku pealt arvutatud veadispersioonide aritmeetiline

Versioon 2.0
Kalev Pärna 31.05.2024



keskmise ja teine on hektaritagavara 20 erineva punkthinnangu pealt arvutatud dispersioon. Lühidalt on see esitatav alljärgneva programmireana

```
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp)),
```

kus m on korduste arv ($m=20$).

3.3.2. Kood

Kõigepealt loeme sisse andmed ja defineerime kasutatavad funktsioonid.

```
setwd(paste0("C:/users/",Sys.getenv("USERNAME"),"/ONEDRI~1/smi_projekt/jatkuprojekt"))

load("uued_smi_andmed.RData")
library(tidyverse)

## — Attaching packages ————— tidyverse 1.3.2 —
## ✓ ggplot2 3.3.6   ✓ purrr 0.3.4
## ✓ tibble 3.1.8   ✓ dplyr 1.0.9
## ✓ tidyr 1.2.0    ✓ stringr 1.4.0
## ✓ readr 2.1.2    ✓ forcats 0.5.1
## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()   masks stats::lag()

#library(tidymodels)
library(mice) #mitmese imputeerimise vahendid

## Warning: package 'mice' was built under R version 4.2.3

##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##   filter
##
## The following objects are masked from 'package:base':
##
##   cbind, rbind

library(ggplot2)
aasta1=2017;aasta2=2021
tmp_prt=prt%>%dplyr::filter(aasta>=aasta1,aasta<=aasta2,psees=="J")
tmp_prtosa_orig=tmp_prt%>%dplyr::select(aproovitykk_id)%>%inner_join(prtosa,by="aproovitykk_id")%>%filter(o
n_sees=="J")
#tmp_klupp=tmp_prt%>%dplyr::select(aproovitykk_id)%>%inner_join(klupp)%>%filter(kaugus<=100,arv_d13>=8)
tmp_trakt=trakt%>%filter(aasta>=aasta1,aasta<=aasta2)
```

Kasutame juba varem (eelmises projektis) välja pakutud meetodit vaatluste teisendamiseks regulaarsele võrgule.

Versioon 2.0
Kalev Pärna 31.05.2024



```
theta=0 #Kui palju on regulaarne võrk pööratud horisontaaltelje suhtes
kyljepikkus=5000 #meetrites, regulaarse võrgustiku ühe ruudu küljepikkus
vec=c(cos(theta),sin(theta))
vec2=c(-sin(theta),cos(theta))
M=matrix(c(vec,vec2),byrow=TRUE,nrow=2)
abi_nihe=tmp_prt%>%pivot_wider(names_from = c(pkylg,pkaugus),values_from=c(koord_e,koord_n),id_cols =atrakt_id)%>%
  mutate(v1x=(koord_e_N_08-koord_e_W_08)/2,v1y=(koord_n_N_08-koord_n_W_08)/2,v2x=(koord_e_N_08-koord_e_E_08)/2,v2y=(koord_n_N_08-koord_n_E_08)/2)
vecs=abi_nihe %>%summarize(across(v1x:v2y,~mean(x,na.rm=TRUE)))
v1=c(vecs$v1x[1],vecs$v1y[1])
v2=c(vecs$v2x[1],vecs$v2y[1])
teisendused=tibble(pkylg=rep(c("W","N","E","S"),each=4),pkaugus=rep(c("02","04","06","08"),times=4),
  muut_x=rep(c(v1[1],-v2[1],-v1[1],v2[1]),each=4)+rep(c(0.5,0,-0.5,-1),times=4)*rep(c(v2[1],v1[1],-v2[1],-v1[1]),each=4),
  muut_y=rep(c(v1[2],-v2[2],-v1[2],v2[2]),each=4)+rep(c(0.5,0,-0.5,-1),times=4)*rep(c(v2[2],v1[2],-v2[2],-v1[2]),each=4))
trakt_keskpunktiga=tmp_prt%>%left_join(teisendused,by=c("pkylg","pkaugus"))%>%group_by(atrakt_id)%>%filter(row_number()==1)%>%
  ungroup()%>%mutate(koord_e=koord_e+muut_x,koord_n=koord_n+muut_y)%>%mutate(id=row_number())%>%
  plyr::select(atrakt_id,koord_e,koord_n,id)
teisendatud=as.matrix(trakt_keskpunktiga%>%dplyr::select(koord_e,koord_n))%*%t(M)
colnames(teisendatud)=c("x","y")
trakt_teisendatud=trakt_keskpunktiga%>%dplyr::bind_cols(as_tibble(teisendatud))

xlimits=range(trakt_teisendatud$x)
ylimits=range(trakt_teisendatud$y)

xp=seq(xlimits[1]-2.5*kyljepikkus,xlimits[2]+2.5*kyljepikkus,by=kyljepikkus)
yp=seq(ylimits[1]-2.5*kyljepikkus,ylimits[2]+2.5*kyljepikkus,by=kyljepikkus)
i_x=as.numeric(cut(trakt_teisendatud$x,breaks=xp))
i_y=as.numeric(cut(trakt_teisendatud$y,breaks=yp))
trakt_piirkond=mutate(trakt_teisendatud,i=i_x,j=i_y)%>%dplyr::select(atrakt_id,i,j)
koik_indeksid=tibble(i=rep(1:max(i_x),each=max(i_y)),j=rep(1:max(i_y),times=max(i_x)))
```

Vaatleme imputeerimist metsamaa tükkidel.

```
m=20 #mitu komplekti andmetest tekitada
seed=20240131 #juhuslikkuse fikseerimine korratavuse huvides
tmp_prtosa_mm <- tmp_prtosa_orig |>filter(maakategooria %in% c("M","MM"))
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm|>dplyr::select(aproovitykk_id,aproovitykkosa_id,prt_aid,aasta,onsaar,qkaitsepohjus,majandkategooria,qmaakond,qomand_kood,maakategooria,maakatgrp,maakatsgrp,kasvukohatyyp,kui_vendus,mulla_tyyp,vanus,inv_korgus,arenguklass,arv_maht_es,peapuuliik,prtliik,inv_diameeter) #fikseerime mingi loet elu tunnustest, mida imputeerimisel kasutada
#tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm|>dplyr::select(aproovitykk_id,aproovitykkosa_id,prt_aid,kui_vendus,arv_maht_es,prtliik)
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm_vahendatud |>mutate(kui_vendus_K=if_else(kui_vendus %in% 'K',1,0),kui_vendus_U=if_else(kui_vendus %in% 'U',1,0),arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("2","6"),NA_real_,arv_maht_es))|>dplyr::select(-kui_vendus)|>dplyr::select(-prtliik,-prt_aid)
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm_vahendatud |>inner_join(tmp_prt|>dplyr::select(aproovitykk_id,koord_n,koord_e))
## Joining, by = "aproovitykk_id"
```

```
prtosa_imp <- mice(tmp_prtosa_mm_vahendatud,m=1,method="rf",seed=seed,maxit=1)

##
## iter imp variable
## 1 1 arv_maht_es inv_diameeter

predMat<-prtosa_imp$predictorMatrix
prtosa_imp <- mice(tmp_prtosa_mm_vahendatud,m=m,method="rf",seed=seed,predictorMatrix = predMat,maxit=5)

predMat["arv_maht_es",c("aproovitykk_id","aproovitykkosa_id")]<- 0
prtosa_long=complete(prtosa_imp,action="long")
#tmp_prtosa_orig=tmp_prtosa
prtosa_long_min=prtosa_long|>dplyr::select(.imp,aproovitykkosa_id,arv_maht_es)|>rename(arv_maht_es_imp=arv_maht_es)
```

Defineerime vajalikud abifunktsioonid sh veaarvutuse 'soome' meetodil.

```
arvuta_sd3=function(andmestik,kv){
  kohandatud=andmestik%>%mutate(hinnang=tinglik_kv-pind_sees*kv)%>%dplyr::select(i,j,hinnang)
  kohandatud2=kohandatud%>%mutate(i=i+1,h2=hinnang)%>%dplyr::select(i,j,h2)
  kohandatud3=kohandatud%>%mutate(i=i+1,j=j+1,h3=hinnang)%>%dplyr::select(i,j,h3)
  kohandatud4=kohandatud%>%mutate(j=j+1,h4=hinnang)%>%dplyr::select(i,j,h4)
  vead=kohandatud%>%
  inner_join(kohandatud2,by=c("i","j"))%>%
  inner_join(kohandatud3,by=c("i","j"))%>%
  inner_join(kohandatud4,by=c("i","j"))%>%
  mutate(vead=(hinnang-h2+h3-h4)/2)%>%pull(vead)
  return(sqrt(sum(vead^2))/sum(andmestik$pind_sees))
}

arvuta_tingimustel=function(cond_A,cond_y,y_string,pind_A,max_per_trakt=16){
  #cond_A on tmp_prtosa jaoks arvesse tulevate proovitükkide tingimus
  #cond_y on tingimus, mis määrab, kas summeeritav suurus nullitakse või mitte
  #y_string annab summeeritava väärtuse veerunime stringina
  #samm on selleks, et saaks hõrendatud võrgul korduvaid arvutusi teha
  #pind_A on tingimusele A vastava Eesti osa teadaolev pindala
  #max_per_trakt annab, mitu proovitükki trakti kohta maksimaalselt arvesse läheb
  taiendatud_trakt=tmp_prtosa%>%mutate(mm_osak=if_else(cond_y,get({y_string}),0),pind_osak=if_else(cond_A,pind_osak,0))%>%
  group_by(atrakt_id)%>%summarize(mm_proc=sum(mm_osak)/max_per_trakt,pind_sees=sum(pind_osak)/max_per_trakt)
  piirkond=taiendatud_trakt%>%inner_join(trakt_piirkond,by="atrakt_id")%>%group_by(i,j)%>%summarize_at(vars(mm_proc,pind_sees),.funs=~mean(.x))%>%ungroup()
  #laiendame nullidega
  piirkond_laiendatud=koik_indeksid%>%left_join(piirkond,by=c("i","j"))%>%mutate(tinglik_kv=coalesce(mm_proc,0),pind_sees=coalesce(pind_sees,0))%>%select(i,j,tinglik_kv,pind_sees)
  tulem=tibble(mu=rep(NA,1),sd=rep(NA,1),#sd2=rep(NA,mitu))
  osakaal=sum(piirkond_laiendatud$tinglik_kv)/sum(piirkond_laiendatud$pind_sees)
  tulem$mu[1]=osakaal*pind_A
  sd3_tul=arvuta_sd3(piirkond_laiendatud,osakaal)
  #tulem$sd[rida]=sd3_tul$sd*pind_tykk*samm^2
  tulem$sd[1]=sd3_tul*pind_A
  return(tulem)
}
```

3.4. Mitmese imputeerimise rakendus: hektaritagavara hindamine kogu metsamaal

Arvutame keskmise hektaritagavara hinnangu koos vastava veahinnanguga imputeeritud andmete baasil, kasutades mitmese imputeerimise meetodikat.

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus kasvukoha proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovity
kkosa_id")|>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("4","8"),arv_maht_es,arv_maht_es_imp))|>mutate(
maht=arv_maht_es*pindosak)
  cond_A=tmp_prtosa$maakategooria %in% c("M","MM")
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,16)

  kv_imp[i]=tmp$mu[1]

  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
#arvutame punkthinnangu
hinnang_imp=mean(kv_imp)
paste0("Imputeerimise abil leitud hinnang: ",hinnang_imp," , hinnangu standardhälve: ",viga_imp)

## [1] "Imputeerimise abil leitud hinnang: 203.943787670853, hinnangu standardhälve: 1.99996750911382"
```

Võrdluseks leiame sama meetodiga keskmise hektaritagavara metsamaal, kasutades üksnes tagavaraproovitükke (st ignoreerides täielikult kasvukoha tükke).

```
taiendatud_trakt=tmp_prtosa_orig%>%inner_join(tmp_prt%>%dplyr::select(prt_aid,pkaugus),by="prt_aid")%>%mut
ate(tagavara=if_else(maakategooria %in% c("M","MM"),arv_maht_es*pindosak,0),pindosak=if_else(maakategooria %i
n% c("M","MM"),pindosak,0))%>%dplyr::filter( pkaugus %in% c("04","08"))%>%
  group_by(atrakt_id)%>%summarize(tagavara=sum(tagavara)/8,pind_sees=sum(pindosak)/8)
piirkond=taiendatud_trakt%>%inner_join(trakt_piirkond,by="atrakt_id")%>%group_by(i,j)%>%summarize_at(vars(t
agavara,pind_sees),.funs=~mean(.x))%>%ungroup()
#laiendame nullidega
piirkond_laiendatud=koik_indeksid%>%left_join(piirkond,by=c("i","j"))%>%mutate(tinglik_kv=coalesce(tagavara,0),
pind_sees=coalesce(pind_sees,0))%>%dplyr::select(i,j,tinglik_kv,pind_sees)
k_tagavara=sum(piirkond_laiendatud$tinglik_kv)/sum(piirkond_laiendatud$pind_sees)
paste0("Keskmine hektaritagavara: ",k_tagavara," , hinnangu standardhälve: ",arvuta_sd3(piirkond_laiendatud,k_tagav
ara))

## [1] "Keskmine hektaritagavara: 203.811889518628, hinnangu standardhälve: 2.24767911821177"
```

Tulemus: Näeme, et kasvukohatükkide täielikul ignoreerimisel keskmise hektaritagavara hinnang jäi praktiliselt samaks (enne 203.9, nüüd 203.8 tm), kuid veahinnang kasvas ca 12% (enne 1.999, nüüd 2.247). Seega kasvukohatükkide kaasamine hektaritagavara hindamisse suurendab hinnangu täpsust 12%, mis on oluline hinnangu paranemine.

3.5. Mitmese imputeerimise rakendus: hektaritagavara hindamine väikeses osakogumis

Püüame kindlaks teha, kas mitmese imputeerimise meetod on rakendatav ka väikeste osakogumite hindamisel.

Näide 1. Esimese boniteedi range kaitsega männikud. Keskmise hektaritagavara

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus kasvukoha proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovity
kkosa_id")>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("4","8"),arv_maht_es,arv_maht_es_imp))>mutate(
maht=arv_maht_es*pindosak)
  cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "MA") & (tmp_prtosa$maja
ndkategooria==2)
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,16)

  kv_imp[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
hinnang_imp=mean(kv_imp)
paste0("Imputeerimise abil leitud hinnang: ",hinnang_imp," , hinnangu standardhälve: ",viga_imp)

## [1] "Imputeerimise abil leitud hinnang: 238.915645127126, hinnangu standardhälve: 7.52010293897976"
```

Võrdluseks teeme arvutused ainult tagavara proovitükkidelt.

```
tmp_prtosa=tmp_prtosa_orig|>filter(str_sub(prt_aid,-1)%in%c("4","8"))>mutate(maht=arv_maht_es*pindosak)
cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "MA") & (tmp_prtosa$majan
dkategooria==2)
tmp=arvuta_tingimustel(cond_A,cond_y=cond_A,"maht",1,8)
paste0("Tagavara proovitükkide abil leitud hinnang: ",tmp$mu," , hinnangu standardhälve: ",tmp$sd)

## [1] "Tagavara proovitükkide abil leitud hinnang: 247.613637063994, hinnangu standardhälve: 8.0696216871048
8"
```

Tulemus (Näide 1): Piirdudes üksnes tagavaraproovitükkidega, saime keskmise hektaritagavara suurema kui kõiki proovitükke kasutades (vastavalt 247.6 ja 238.9tm/h), kuid see erinevus jääb arvutatud vea piiresse. Samas on näha, et kasvukohaproovitükkidele tagavarade imputeerimine suurendab hinnangu täpsust ca 7% (standardhälbe vähenemine 8,07 pealt 7.52 -ni).

Näide 2. Esimese boniteedi range kaitsega kuusikud. Keskmine hektaritagavara

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus kasvukoha proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovity
kkosa_id")|>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("4","8"),arv_maht_es,arv_maht_es_imp))|>mutate(
maht=arv_maht_es*pindosak)
  cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "KU") & (tmp_prtosa$majand
dkategooria==2)
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,16)

  kv_imp[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
hinnang_imp=mean(kv_imp)
paste0("Imputeerimise abil leitud hinnang: ",hinnang_imp," , hinnangu standardhälve: ",viga_imp)

## [1] "Imputeerimise abil leitud hinnang: 297.137252627254, hinnangu standardhälve: 10.9989790589468"
```

Võrdluseks teeme arvutused ainult tagavara proovitükkidelt.

```
tmp_prtosa=tmp_prtosa_orig|>filter(str_sub(prt_aid,-1)%in%c("4","8"))|>mutate(maht=arv_maht_es*pindosak)
cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "KU") & (tmp_prtosa$majand
kategooria==2)
tmp=arvuta_tingimustel(cond_A,cond_y=cond_A,"maht",1,8)
paste0("Tagavara proovitükkide abil leitud hinnang: ",tmp$mu," , hinnangu standardhälve: ",tmp$sd)

## [1] "Tagavara proovitükkide abil leitud hinnang: 306.565593026968, hinnangu standardhälve: 11.340612398680
6"
```

Tulemus (Näide 2): Piirdudes üksnes tagavaraproovitükkidega, saime keskmise hektaritagavara suurema kui kõiki proovitükke kasutades (vastavalt 297.1 ja 306.6tm/h), kuid see erinevus jääb arvatud vea piirsesse. Samas on näha, et kasvukohaproovitükkidele tagavarade imputeerimine suurendab hinnangu täpsust ca 3% (standardhälbed enne ja pärast imputeerimist vastavalt 11.3 ja 11.0).

Kokkuvõte: Ülaltoodud analüüs näitab, et kasvukohaproovitükkide kaasamine keskmise tagavara hindamisse mitmese imputeerimise meetodil suurendab hinnangu täpsust ca 12% võrra. Väikestel osakogumitel oli täpsuse suurenemine tagasihoidlikum (3-7%), olenevalt konkreetsest olukorrast.

3.6. Relaskoobiandmete vajalikkuse analüüs kasvukohaproovitükkidel

3.6.1. Sissejuhatus

Eespool tegime SMI andmete põhjal kindlaks, et kasvukohaproovitükkidele mahtude prognoosimine mitmese imputeerimise meetodil, kasutades seal juba olemasolevaid tunnuseid (pikk tunnuste nimekiri, kus pole relaskoobinäitu), suurendab kogutagavara hinnangu täpsust ca 12% ja on selles mõttes õigustatud.

Samas on üks meie varasem analüüs (mida me siinkohal pikemalt ei käsitle) näidanud, et relaskoobinäidu lisamine praeguse mahumudeli argumenttunnuste komplekti (H100, onsaar, vanus) suurendaks tagavarahinnangu täpsust ca 8%.

Seega lahtiseks jääb küsimus, kas relaskoobinäidu kasutegur (8%) jääb alles ka siis, kui ta lisada juba niigi pikka imputeerimisel kasutatavasse argumenttunnuste nimekirja.

3.6.2. Analüüsimeetod ja kood

Selleks, et relaskoobinäidu kasutegurit hinnata, mängime läbi situatsiooni, kus meil on kasutada **mõõtmistulemused ainult trakti nurkades** ning uurime külgede keskel olevate, koodiga 04 tagavaraproovitükkide mahtude (mis tegelikult on teada) imputeerimise efekti. Külje keskkoha mahtu prognoosime kahel erineval viisil: 1) ühel juhul kasutame imputeerimisel ainult selliseid andmeid, mis kasvukohaproovitükkidel on tavapäraselt teada (sinna ei kuulu relaskoobinäit); 2) teisel juhul võtame lisaks appi ka vaadeldava proovitüki relaskoobiandmed (mis proovitükkidel 04 on olemas).

Mahtude prognoosimiseks kasutame mitmese imputeerimise tehnikat, kusjuures puuduvate väärtuste asendamiseks kasutame juhumetsa (Random Forests) meetodit.

Imputeerimise tulemusel tekkinud 'täielike' mahuandmete (mis antud simulatsioonis sisaldavad üksnes tagavaraproovitükke) põhjal arvutame kogutagavara hinnangu koos selle standardhälbega. Standardhälbe leidmiseks kasutame (nagu ka eespool) nn 'soome' meetodit.

Kood

Alljärgnev kood mõnevõrra kordab ülaltoodud koodi, kuid ülevaatlikkuse huvides ei ole seda lühendatud.

Kõigepealt loeme sisse andmed ja defineerime kasutatavad funktsioonid.

```
setwd(paste0("C:/users/",Sys.getenv("USERNAME"),"/ONEDRI~1/smi_projekt/jatkuprojekt"))
```

Version 2.0
Kalev Pärna 31.05.2024



```
load("uued_smi_andmed.RData")
library(tidyverse)

## — Attaching packages — tidyverse 1.3.2 —
## ✓ ggplot2 3.3.6   ✓ purrr 0.3.4
## ✓ tibble 3.1.8   ✓ dplyr 1.0.9
## ✓ tidyr 1.2.0    ✓ stringr 1.4.0
## ✓ readr 2.1.2   ✓ forcats 0.5.1
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()   masks stats::lag()

#library(tidymodels)
library(mice) #mitmese imputeerimise vahendid

## Warning: package 'mice' was built under R version 4.2.3

##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
## filter
##
## The following objects are masked from 'package:base':
##
## cbind, rbind

library(ggplot2)
aasta1=2017;aasta2=2021
tmp_prt=prt%>%dplyr::filter(aasta>=aasta1,aasta<=aasta2,psees=="J")
tmp_prtosa_orig=tmp_prt%>%dplyr::select(aproovitykk_id)%>%inner_join(prtosa,by="aproovitykk_id")%>%filter(o
n_sees=="J")
tmp_trakt=trakt%>%filter(aasta>=aasta1,aasta<=aasta2)
```

Kasutame varempakutud meetodit vaatluste teisendamiseks regulaarsele võrgule.

```
theta=0 #Kui palju on regulaarne võrk pööratud horisontaaltelje suhtes
kyljepikkus=5000 #meetrites, regulaarse võrgustiku ühe ruudu küljepikkus
vec=c(cos(theta),sin(theta))
vec2=c(-sin(theta),cos(theta))
M=matrix(c(vec,vec2),byrow=TRUE,nrow=2)
abi_nihe=tmp_prt%>%pivot_wider(names_from = c(pkylg,pkaugus),values_from=c(koord_e,koord_n),id_cols =atrakt_i
d)%>%
  mutate(v1x=(koord_e_N_08-koord_e_W_08)/2,v1y=(koord_n_N_08-koord_n_W_08)/2,v2x=(koord_e_N_08-koord_e_E
_08)/2,v2y=(koord_n_N_08-koord_n_E_08)/2)
vecs=abi_nihe %>%summarize(across(v1x:v2y,~mean(x,na.rm=TRUE)))
v1=c(vecs$v1x[1],vecs$v1y[1])
v2=c(vecs$v2x[1],vecs$v2y[1])
teisendused=tibble(pkylg=rep(c("W","N","E","S"),each=4),pkaugus=rep(c("02","04","06","08"),times=4),
  muut_x=rep(c(v1[1],-v2[1],-v1[1],v2[1]),each=4)+rep(c(0.5,0,-0.5,-1),times=4)*rep(c(v2[1],v1[1],-v2[1],-v1
[1]),each=4),
  muut_y=rep(c(v1[2],-v2[2],-v1[2],v2[2]),each=4)+rep(c(0.5,0,-0.5,-1),times=4)*rep(c(v2[2],v1[2],-v2[2],-v1
[2]),each=4))
```

```
trakt_keskpunktiga=tmp_prt%>%left_join(teisendused,by=c("pkylg","pkaugus"))%>%group_by(atrakt_id)%>%filter(
row_number()==1)%>%
  ungroup()%>%mutate(koord_e=koord_e+muut_x,koord_n=koord_n+muut_y)%>%mutate(id=row_number())%>%d
plyr::select(atrakt_id,koord_e,koord_n,id)
teisendatud=as.matrix(trakt_keskpunktiga%>%dplyr::select(koord_e,koord_n))%*%t(M)
colnames(teisendatud)=c("x","y")
trakt_teisendatud=trakt_keskpunktiga%>%dplyr::bind_cols(as_tibble(teisendatud))

xlimits=range(trakt_teisendatud$x)
ylimits=range(trakt_teisendatud$y)

xp=seq(xlimits[1]-2.5*kyljepikkus,xlimits[2]+2.5*kyljepikkus,by=kyljepikkus)
yp=seq(ylimits[1]-2.5*kyljepikkus,ylimits[2]+2.5*kyljepikkus,by=kyljepikkus)
i_x=as.numeric(cut(trakt_teisendatud$x,breaks=xp))
i_y=as.numeric(cut(trakt_teisendatud$y,breaks=yp))
trakt_piirkond=mutate(trakt_teisendatud,i=i_x,j=i_y)%>%dplyr::select(atrakt_id,i,j)
koik_indeksid=tibble(i=rep(1:max(i_x),each=max(i_y)),j=rep(1:max(i_y),times=max(i_x)))
```

Vaatleme imputeerimist metsamaa tagavaraproovitükkidel.

```
m=20 #mitu andmekomplekti tekitada
seed=20240220 #juhuslikkuse fikseerimine korratavuse huvides
#vaatleme ainult tagavara proovitükke
tmp_prtosa_mm <- tmp_prtosa_orig |>filter(maakategooria %in% c("M","MM"),str_sub(prt_aid,-1)%in%c("4","8"))
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm|>dplyr::select(aproovitykk_id,aproovitykkosa_id,prt_aid,aasta,onsaar
,qkaitsepohjus,majandkateooria,qmaakond,qomand_kood,maakategooria,maakatgrp,maakatsgrp,kasvukohatyp,kui
vendus,mulla_tyyp,vanus,inv_korgus,arenguklass,arv_maht_es,peapuuliik,inv_diameeter,inv_g1,inv_g2) #fikseerime mi
ngi loetelu tunnustest, mida imputeerimisel kasutada
#tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm|>dplyr::select(aproovitykk_id,aproovitykkosa_id,prt_aid,kuivendus,arv_
maht_es,prtliik)
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm_vahendatud |>mutate(kuivendus_K=if_else(kuivendus %in% 'K',1,0),
kuivendus_U=if_else(kuivendus %in% 'U',1,0),arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("4"),NA_real_,arv_maht_
es),relaskoop=coalesce((inv_g1+inv_g2)/2,inv_g1,inv_g2))|>dplyr::select(-kuivendus)|>dplyr::select(-prt_aid)|>select
(-inv_g1,-inv_g2)
tmp_prtosa_mm_vahendatud <- tmp_prtosa_mm_vahendatud |>inner_join(tmp_prt|>dplyr::select(aproovitykk_id,koor
d_n,koord_e))

## Joining, by = "aproovitykk_id"

prtosa_imp <- mice(tmp_prtosa_mm_vahendatud,m=1,method="rf",seed=seed,maxit=1,printFlag = FALSE)
predMat<-prtosa_imp$predictorMatrix
predMat["arv_maht_es",c("aproovitykk_id","aproovitykkosa_id")]<- 0
prtosa_imp <- mice(tmp_prtosa_mm_vahendatud,m=m,method="rf",seed=seed,predictorMatrix = predMat,maxit=5,pr
intFlag = FALSE)
prtosa_long=complete(prtosa_imp,action="long")
#tmp_prtosa_orig=tmp_prtosa
prtosa_long_min=prtosa_long|>dplyr::select(imp,aproovitykkosa_id,arv_maht_es)|>rename(arv_maht_es_imp=arv_ma
ht_es)

#Ilma relaskoobita
print("Imputeerimine ilma relaskoobita algas")

## [1] "Imputeerimine ilma relaskoobita algas"
```

```
tmp_prtosa_mm_vahendatud2 <- select(tmp_prtosa_mm_vahendatud,-relaskoop)
prtosa_imp2 <- mice(tmp_prtosa_mm_vahendatud2,m=1,method="rf",seed=seed,maxit=1,printFlag = FALSE)
predMat2<-prtosa_imp2$predictorMatrix
predMat2["arv_maht_es",c("aproovitykk_id","aproovitykkosa_id")]<- 0
prtosa_imp2 <- mice(tmp_prtosa_mm_vahendatud2,m=m,method="rf",seed=seed,predictorMatrix = predMat2,maxit=
5,printFlag = FALSE)
prtosa_long2=complete(prtosa_imp2,action="long")
#tmp_prtosa_orig=tmp_prtosa
prtosa_long_min2=prtosa_long2|>dplyr::select(.imp,aproovitykkosa_id,arv_maht_es)|>rename(arv_maht_es_imp=arv_
maht_es)
```

Defineerime abifunktsioonid

```
arvuta_sd3=function(andmestik,kv){
kohandatud=andmestik%>%mutate(hinnang=tinglik_kv-pind_sees*kv)%>%dplyr::select(i,j,hinnang)
kohandatud2=kohandatud%>%mutate(i=i+1,h2=hinnang)%>%dplyr::select(i,j,h2)
kohandatud3=kohandatud%>%mutate(i=i+1,j=j+1,h3=hinnang)%>%dplyr::select(i,j,h3)
kohandatud4=kohandatud%>%mutate(j=j+1,h4=hinnang)%>%dplyr::select(i,j,h4)
vead=kohandatud%>%
inner_join(kohandatud2,by=c("i","j"))%>%
inner_join(kohandatud3,by=c("i","j"))%>%
inner_join(kohandatud4,by=c("i","j"))%>%
mutate(vead=(hinnang-h2+h3-h4)/2)%>%pull(vead)
return(sqrt(sum(vead^2))/sum(andmestik$pind_sees))
}

arvuta_tingimustel=function(cond_A,cond_y,y_string,pind_A,max_per_trakt=8){
#cond_A on tmp_prtosa jaoks arvessetulevate proovitükkide tingimus
#cond_y on tingimus, mis määrab, kas summeeritav suurus nullitakse või mitte
#y_string annab summeeritava väärtuse veerunime stringina
#samm on selleks, et saaks hõrendatud võrgul korduvaid arvutusi teha
#pind_A on tingimusele A vastava Eesti osa teadaolev pindala
#max_per_trakt annab, mitu proovitükki trakti kohta maksimaalselt arvesse läheb
taiendatud_trakt=tmp_prtosa%>%mutate(mm_osak=if_else(cond_y,get({y_string}),0),pind_osak=if_else(cond_A,pi
ndosak,0))%>%
group_by(atrakt_id)%>%summarize(mm_proc=sum(mm_osak)/max_per_trakt,pind_sees=sum(pind_osak)/max_per
_trakt)
piirkond=taiendatud_trakt%>%inner_join(trakt_piirkond,by="atrakt_id")%>%group_by(i,j)%>%summarize_at(vars
(mm_proc,pind_sees),.funs=~mean(.x))%>%ungroup()
#laiendame nullidega
piirkond_laiendatud=koik_indeksid%>%left_join(piirkond,by=c("i","j"))%>%mutate(tinglik_kv=coalesce(mm_proc,0
),pind_sees=coalesce(pind_sees,0))%>%select(i,j,tinglik_kv,pind_sees)
tulem=tibble(mu=rep(NA,1),sd=rep(NA,1))#sd2=rep(NA,mitu))
osakaal=sum(piirkond_laiendatud$tinglik_kv)/sum(piirkond_laiendatud$pind_sees)
tulem$mu[1]=osakaal*pind_A
sd3_tul=arvuta_sd3(piirkond_laiendatud,osakaal)
#tulem$sd[rida]=sd3_tul$sd*pind_tykk*samm^2
tulem$sd[1]=sd3_tul*pind_A
return(tulem)
}
```

3.6.3. Hektaritagavara hindamine kogu metsamaal kasutades ka relaskoobiandmeid

Arvutame keskmise hektaritagavara hinnangu koos selle veahinnanguga imputeeritud andmete baasil, kasutades mitmese imputeerimise metoodikat. Esimene arvutus põhineb imputeeritud andmestikel, mille leidmisel **kasutati ka relaskoobiandmeid**.

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus kasvukoha proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovity
kkosa_id")|>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("8"),arv_maht_es,arv_maht_es_imp))|>mutate(ma
ht=arv_maht_es*pindosak)
  cond_A=tmp_prtosa$maakategooria %in% c("M","MM") &str_sub(tmp_prtosa$prt_aid,-1)%in%c("4","8")
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,8)

  kv_imp[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
hinnang_imp=mean(kv_imp)
paste0("Relaskoobi abil imputeerides saadud hinnang: ",hinnang_imp," , hinnangu standardhälve: ",viga_imp)

## [1] "Relaskoobi abil imputeerides saadud hinnang: 206.338581643957, hinnangu standardhälve: 2.48126544905
666"
```

3.6.4. Hektaritagavara hindamine kogu metsamaal ilma relaskoobiandmeteta

Võrdluseks teeme samad arvutused imputeeritud andmestikel, mille leidmise käigus **ei kasutanud relaskoobi andmeid**.

```
kv_imp2=rep(NA,m)
sd_imp2=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus kasvukoha proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min2|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovity
ykkosa_id")|>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("8"),arv_maht_es,arv_maht_es_imp))|>mutate(m
aht=arv_maht_es*pindosak)
  cond_A=tmp_prtosa$maakategooria %in% c("M","MM") &str_sub(tmp_prtosa$prt_aid,-1)%in%c("4","8")
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,8)

  kv_imp2[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp2[i]=tmp$sd[1]
}
viga_imp2=sqrt(mean(sd_imp2^2)+(m+1)/m*var(kv_imp2))
hinnang_imp2=mean(kv_imp2)
paste0("Ilma relaskoobiinfota imputeerides leitud hinnang: ",hinnang_imp2," , hinnangu standardhälve: ",viga_imp2)
```

```
## [1] "Ilma relaskoobiinfota imputeerides leitud hinnang: 206.95107807861, hinnangu standardhälve: 2.5185584354024"
```

3.6.5. Kokkuvõte (relaskoobimõõtmiste vajalikkus kasvukohtadel)

Kokkuvõte: Võrreldes omavahel relaskoobi abiga ja ilma relaskoobita saadud tulemusi, näeme, et keskmise hektaritagavara hinnang on mõlemal juhul praktiliselt sama (vastavalt 206.33 ja 206.95). Väga lähedased on ka hinnangute standardhälbed (2.481 ja 2.518), mis tähendab, et relaskoobiandmed suurendasid keskmise hektaritagavara hinnangu täpsust ainult 1.5%. Ilmselt ei ole see piisav argument relaskoobimõõtmiste rakendamiseks kasvukohatükkidel.

Kommentaari: Ilmselt on saadud tulemuse põhjus selles, et relaskoobiinfo on paljuski kaetud teiste, juba olemas olevate tunnuste poolt. Tegelikult on põhiline osa varieeruvusest traktide vaheline, nii et traktisest hinnangute mõningane täpsustus, mida relaskoobinäidult võiks oodata, ei pruugi anda koguhinnangule kuigi palju juurde.

3.7. Kasvukohaproovitükkide vajalikkuse täiendav testimine tagavara hindamisel

Punktis 3.4 kõiki proovitükke käsitleva analüüsiga leiti, et kasvukohtade kaasamine mitmese imputeerimise abil suurendab hektaritagavara hinnangu täpsust 12%. Püüame sellele väitele saada tuge ka teisiti organiseeritud andmete pealt, mis olid kasutusel juba eelmises punktis (vaatluse all seega ainult tagavaraproovitükid, kusjuures 04 koodiga tükid on mõtteliselt kuulutatud 'kasvukohaproovitükkideks').

Leiame keskmise hektaritagavara metsamaal, kasutades ainult trakti nurkades olevaid tagavara proovitükke. Sel viisil imiteerime olukorda, kus 'kasvukohaproovitükid' (külje keskel asuvad proovitükid koodiga 04) jäävad kogutagavara hindamisest täielikult välja.

```
laiendatud_trakt=tmp_prtosa_orig%>%inner_join(tmp_prt%>%dplyr::select(prt_aid,pkaugus),by="prt_aid")%>%mut
ate(tagavara=if_else(maakategooria %in% c("M","MM"),arv_maht_es*pindosak,0),pindosak=if_else(maakategooria %i
n% c("M","MM"),pindosak,0))%>%dplyr::filter( pkaugus %in% c("08"))%>%
group_by(atrakt_id)%>%summarize(tagavara=sum(tagavara)/4,pind_sees=sum(pindosak)/4)
piirkond=laiendatud_trakt%>%inner_join(trakt_piirkond,by="atrakt_id")%>%group_by(i,j)%>%summarize_at(vars(t
agavara,pind_sees),funs=~mean(.x))%>%ungroup()
#laiendame nullidega
piirkond_laiendatud=koik_indeksid%>%left_join(piirkond,by=c("i","j"))%>%mutate(tinglik_kv=coalesce(tagavara,0),
pind_sees=coalesce(pind_sees,0))%>%dplyr::select(i,j,tinglik_kv,pind_sees)
k_tagavara=sum(piirkond_laiendatud$tinglik_kv)/sum(piirkond_laiendatud$pind_sees)
paste0("Keskmine hektaritagavara: ",k_tagavara," , hinnangu standardhälve: ",arvuta_sd3(piirkond_laiendatud,k_tagav
ara))
## [1] "Keskmine hektaritagavara: 203.329448873467, hinnangu standardhälve: 2.86678521501002"
```

Tulemus: Kasutades üksnes trakti nurkades asuvate tagavaraproovitükkide andmeid (ehk ignoreerides täielikult 04-koodiga 'kasvukohaproovitükkide' andmeid), tuli keskmise hektaritagavara hinnang pisut väiksem võrreldes eespool (punktides 3.6.3 ja 3.6.4) vaadeldud imputeeritud mahtude juhuga (nüüd 203.32 vs enne 206), kuid erinevus jääb selgelt vea piiresse. Samas hinnangu standardhälve tuleb 'kasvukohaproovitükke' kõrvale jättes märgatavalt suurem (2.866 vs enne 2.5), mis tähendab hinnangutäpsuse kahanemist ca 15%.

Kommentaari: Siinne toodud tulemus on heas kooskõlas punktis 3.4 saadud tulemusega, kus ulatuslikuma, kõiki proovitükke haarava analüüsiga leiti, et kasvukohtade kaasamine suurendab hektaritagavara hinnangu täpsust 12%.

3.8. Kasvukohaproovitükkide täiendav testimine väikeste osakogumite hindamisel

Järgnevas püüame punktides 3.6 ja 3.7 kasutatud taktikat (mis põhineb üksnes tagavaraproovitükkide andmetel) rakendades välja selgitada, kas üldkogumi kohta saadud järeldused kehtivad ka väiksemate osakogumite korral.

Näide 1: Esimese boniteedi range kaitsega männikud. Keskmise hektaritagavara

Nagu eespool üldkogumi korral, nii ka nüüd osakogumi korral kasutame oma imitatsiooni-protsessis üksnes tagavaraproovitükkide andmeid, kuna seal on relaskoobiandmed olemas.

3.8.1. Relaskoobinäidu vajalikkuse testimine imputeerimisel (väike osakogum)

Esmalt vaatleme olukorda, kus külje keskel asuvate 'kasvukohaproovitükkide' (kood 04) mahud on **imputeeritud relaskoobinäitu kaasates**.

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus 'kasvukoha' proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovitykkosa_id")>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("8"),arv_maht_es,arv_maht_es_imp))>mutate(maht=arv_maht_es*pindosak)
  cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "MA") & (tmp_prtosa$majandkategooria==2)&str_sub(tmp_prtosa$prt_aid,-1)%in%c("4","8")
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,8)

  kv_imp[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
hinnang_imp=mean(kv_imp)
```


Versioon 2.0
Kalev Pärna 31.05.2024



```
paste0("Relaskoobi tulemust kasutava imputeerimise abil saadud hinnang: ",hinnang_imp,", hinnangu standardhälve: ",viga_imp)
```

```
## [1] "Relaskoobi tulemust kasutava imputeerimise abil saadud hinnang: 248.183858911161, hinnangu standardhälve: 8.83490559247256"
```

Nüüd teeme samad arvutused olukorras, kus **imputeerimisel relaskoobinäitu ei kasutata**:

```
kv_imp=rep(NA,m)
sd_imp=rep(NA,m)
for(i in 1:m){
  #tekitame i-nda andmestiku, kus 'kasvukoha' proovitükkide mahud on imputeeritud
  tmp_prtosa=tmp_prtosa_orig|>left_join(prtosa_long_min2|>dplyr::filter(.imp==i)|>dplyr::select(-.imp),by = "aproovit_ykkosa_id")|>mutate(arv_maht_es=if_else(str_sub(prt_aid,-1)%in%c("8"),arv_maht_es,arv_maht_es_imp))|>mutate(maht=arv_maht_es*pindosak)
  cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "MA") & (tmp_prtosa$majandkategooria==2)&str_sub(tmp_prtosa$prt_aid,-1)%in%c("4","8")
  cond_y=cond_A
  tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,8)

  kv_imp[i]=tmp$mu[1]
  #arvutame standardhälbe hinnangu
  sd_imp[i]=tmp$sd[1]
}
viga_imp=sqrt(mean(sd_imp^2)+(m+1)/m*var(kv_imp))
hinnang_imp=mean(kv_imp)
paste0("Ilma relaskoobi näiduta imputeerimise abil saadud hinnang: ",hinnang_imp,", hinnangu standardhälve: ",viga_imp)

## [1] "Ilma relaskoobi näiduta imputeerimise abil saadud hinnang: 244.829435808697, hinnangu standardhälve: 8.29630054753341"
```

Tulemus: Näeme, et väikese osakogumi hektaritagavara prognoosimisel relaskoobiandmete kasutuselevõttuga hinnangu viga isegi suurenes (8.83 on 6% suurem kui 8.29). See on näiliselt loogikavastane tulemus, sest täiendava info kasutamine prognoosimisel üldiselt ei tohiks prognoositulemust halvendada. Nähtavasti on siin tegemist väikeste andmetega paratamatult kaasnevate suurte hinnanguvigadega.

3.8.2. Kasvukohatükkide vajalikkuse testimine tagavara hindamisel (väike osakogum)

Lõpuks kontrollime seda, kas kasvukoha proovitükkide täielik ignoreerimine väikese osakogumi mahu hindamisel vähendab hinnangu täpsust oluliselt või mitte. Selleks hindame osakogumi mahtu ainult traktinurkade baasil:

```
tmp_prtosa=tmp_prtosa_orig|>mutate(maht=arv_maht_es*pindosak)
cond_A=(tmp_prtosa$maakategooria %in% c("M","MM")) & (tmp_prtosa$arv_epl %in% "MA") & (tmp_prtosa$majandkategooria==2)&str_sub(tmp_prtosa$prt_aid,-1)%in%c("8")
cond_y=cond_A
tmp=arvuta_tingimustel(cond_A,cond_y,"maht",1,4)
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
kv_imp=tmp$mu[1]
#arvutame standardhälbe hinnangu
sd_imp=tmp$sd[1]
paste0("Ainult nurkades olevate proovitükkide abil saadud hinnang: ",kv_imp,", hinnangu standardhälve: ",sd_imp)

## [1] "Ainult nurkades olevate proovitükkide abil saadud hinnang: 249.050912930651, hinnangu standardhälve: 9.84225867920885"
```

Tulemus: Kasvukohaproovitükkide täielik ignoreerimine väikeste osakogumite mahu hinda-misel suurendas hinnanguviga 15% (9.84 vs $8.56=(8.29+8.83)/2$), mis ühtib sama näitajaga kogumahu hindamisel (15%).

3.9. Kokkuvõte

Analüüsi tulemusena selgus, et nii üldkogumi kui ka väiksema osakogumi mahu hindamisel on otstarbekas kasutada ka kasvukohaproovitükkide andmeid, mille kaasamine suurendab hektaritagavara hinnangu täpsust ca 12-15%. Seejuures näidati, et sobiv meetodika kasvukohatükkidele mahtude prognoosimiseks on mitmese imputeerimise meetod. Uus meetod võimaldab muu hulgas ka korrektset veaarvutust kogutagavara hinnangule (mis oli seni SMI-s kasutatud mahumudelite puudus). Selgus ka, et ei ole vajadust lisada kasvukohtadel mõõdetavate tunnuste hulka Bitterlichi relaskoobi näitu, kuna sealt laekuv info oleks suures osas kaetud teiste, mudelisse juba haaratud tunnuste poolt.

4. Optimaalne ajaaken mudelite hindamiseks

4.1. Sissejuhatus

SMI käivitamisele (1999) järgnenud aja jooksul on looduslikud tingimused varieerunud märgatavalt, sh on toimunud kliimatilised muudatused, mis on oletatavasti avaldanud mõju ka metsa kasvule. Olukord on muutunud ka metsakuivenduse ja põllukuivenduse tõttu, mis oli eriti intensiivne aastail 1960-1990 ja mille mõju kestab siiani.

SMI seisukohast on oluline teada, kas SMI-s kasutatavad mudelid, mis hetkel on hinnatud SMI kogu ajaloo põhjal, vajavad aeg-ajalt ka uuendamist ehk rekaliibreerimist värskemate andmete alusel. Järgneva analüüsiga püütaksegi anda vastus küsimusele, milline on optimaalne ajaaken (andmeintervall), mille põhjal kalibreeritud mudelid annavad täpseima prognoosi. Vaatluse alla võetakse eeskätt puu kõrguse ja hektaritagavara mudelid.

Esmalt kirjeldame, kasutades lihtsat keskmistamist, kui suur on looduslike muutuste (sh metsmajanduslike võtete) koosmõju puude kasvule. Selleks vaatleme puude kõrguse muutumise dünaamikat kogu SMI toimise perioodil.

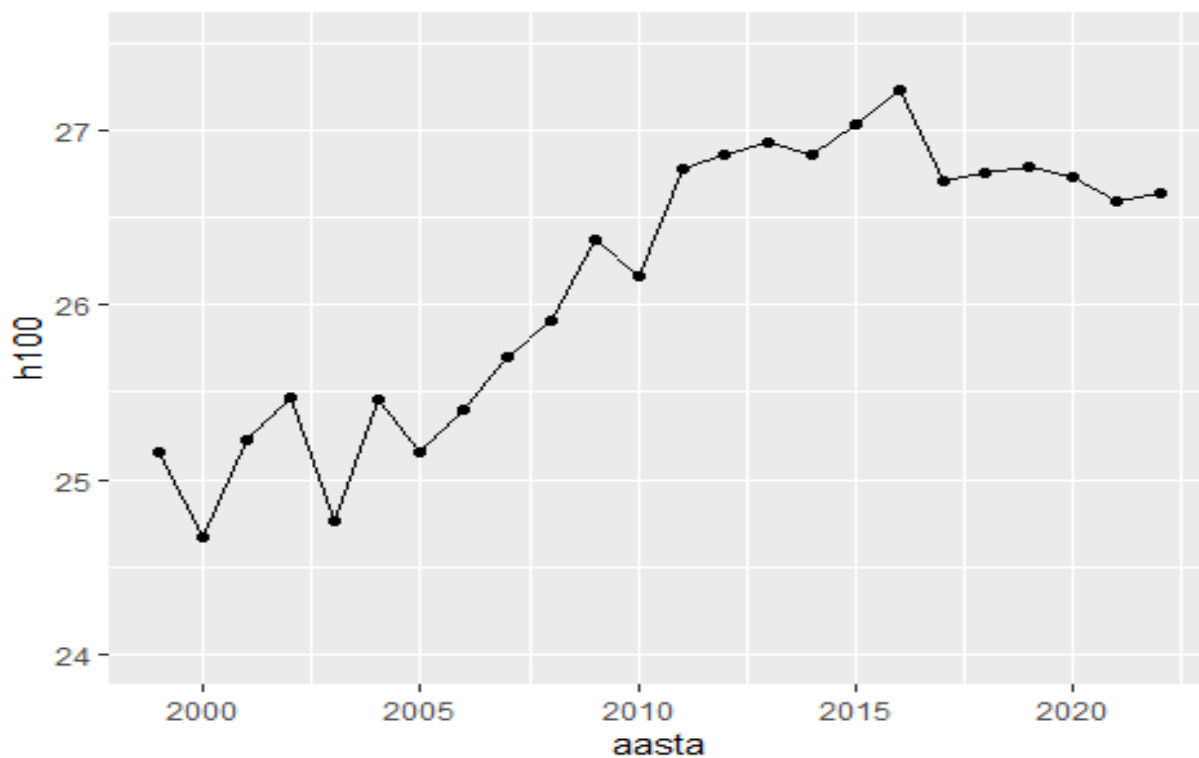
4.2. Eelanalüüs: h100 keskmise muutumine ajas

Eesmärgiks on teha kindlaks, kas looduse käitumises on mingeid tuvastatavaid nihkeid toimunud viimastel aastakümnetel, alates SMI käivitamisest 1999. Metsaekspertide hinnangul on puude kasv selle aja sees kiirenenud. Järgnevas püüame välja selgitada, kuidas H100 ja kõrguse kasv konkreetsemalt avaldub, aga ka seda, kas nimetatud kasv pole hoopiski seotud mõne meetoodilise muutusega 23 aasta jooksul (H100 mudeli võimalik muutus, mõõteprotseduuride jne muutus).

```
korgusandmed <- prtosa |> select(atrakt_id,aasta,inv_korgus,arv_h100,vanus,peapuuliik,maakategooria,arv_epl,prtliik,  
majandkategooria)|>filter(!is.na(arv_h100))
```

- 1) Alustuseks esitame tunnuse H100 (puistu arvutuslik kõrgus 100 aastaselt) keskmiste muutumise dünaamika ajaperioodil 1999-2022.

```
joonisele <- korgusandmed|>filter(maakategooria %in%  
c("M"))|>group_by(aasta)|>summarize(h100=mean(arv_h100))  
ggplot(joonisele,aes(x=aasta,y=h100))+geom_line()+geom_point()+ylim(24,27.5)
```

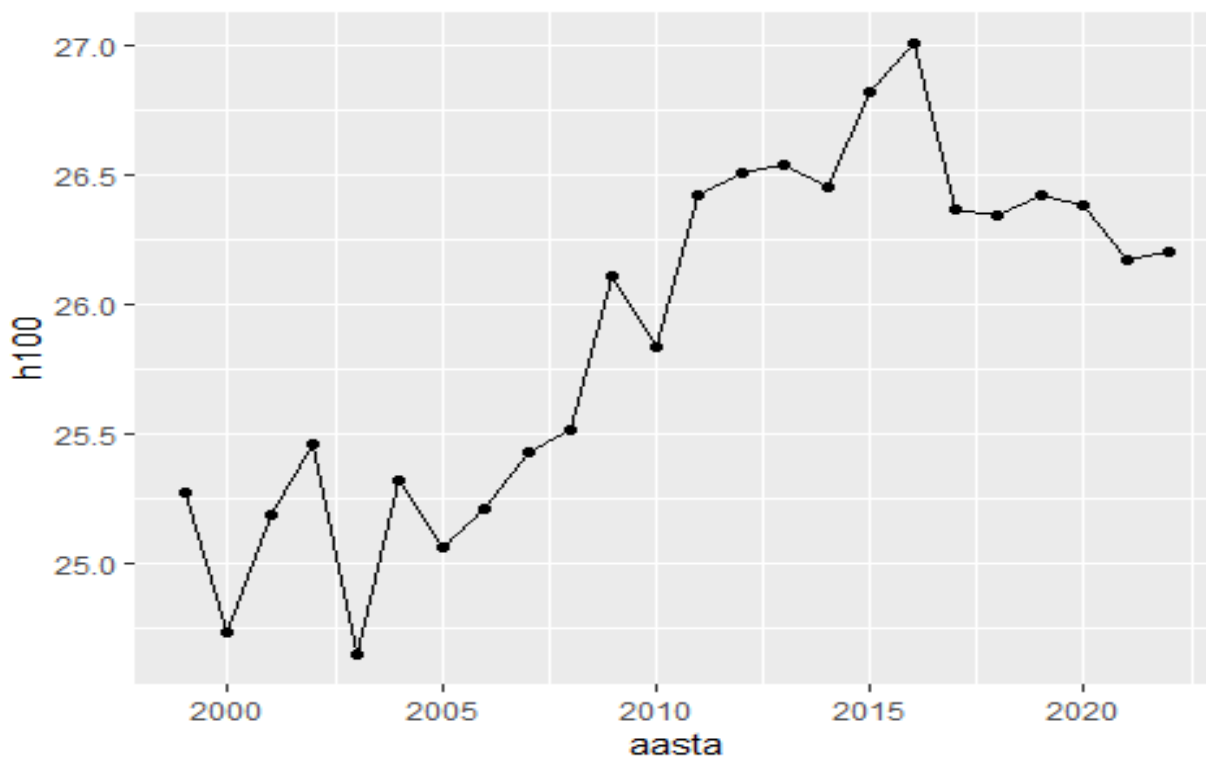


Joonis 3

Tulemus: jooniselt on näha H100 kasv ca 2 m võrra aastatel 2004 -2016 (SMI II ja III periood), millel järgneb teatav langus (ca 0,5m) aastail 2017-2022.

- 2) Võrdluseks vaatleme ainult neid proovitükke, kus H100 on arvutatud mudeli järgi (vanus >15 aastat):

```
joonisele <- korgusandmed|>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1)|>group_by(aasta)|>summarize(h100=mean(arv_h100))  
ggplot(joonisele,aes(x=aasta,y=h100))+geom_line()+geom_point()
```



Joonis 4

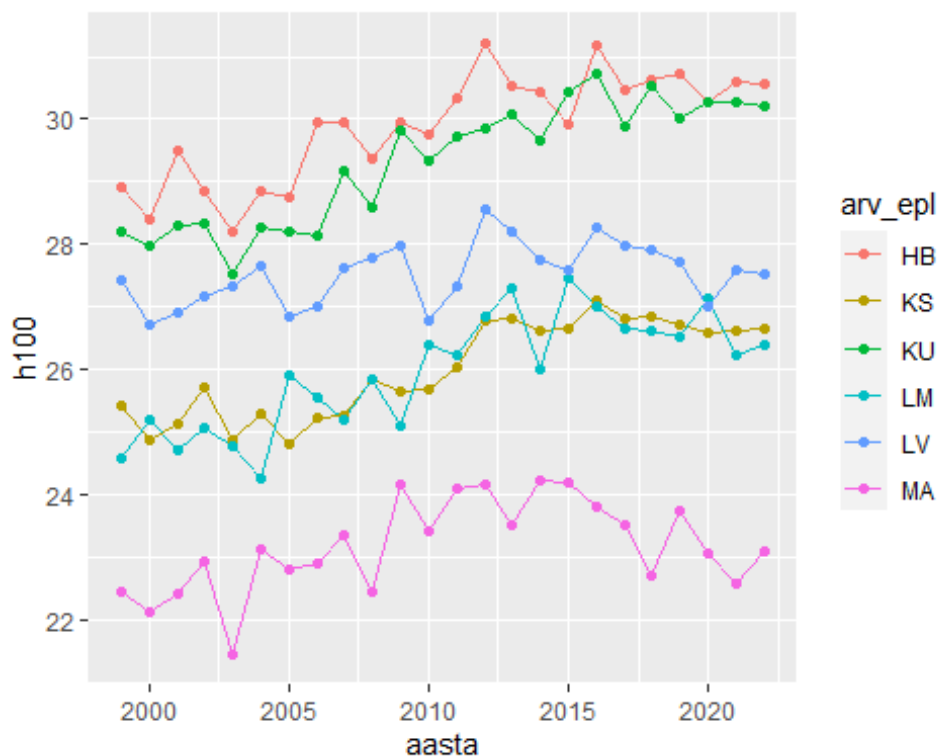
Tulemus: näeme, et pilt on väga sarnane eelnevaga.

3) Sama joonis peapuuliigiti :

```
joonisele <- korgusandmed|>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1,arv_epl!="XX")|>group_by(aasta,arv_epl)|>summarize(h100=mean(arv_h100))
```

```
## `summarise()` has grouped output by 'aasta'. You can override using the ## `.groups` argument.
```

```
ggplot(joonisele,aes(x=aasta,y=h100))+geom_line(aes(group=arv_epl,color=arv_epl))+geom_point(aes(group=arv_epl,color=arv_epl))
```



Joonis 5

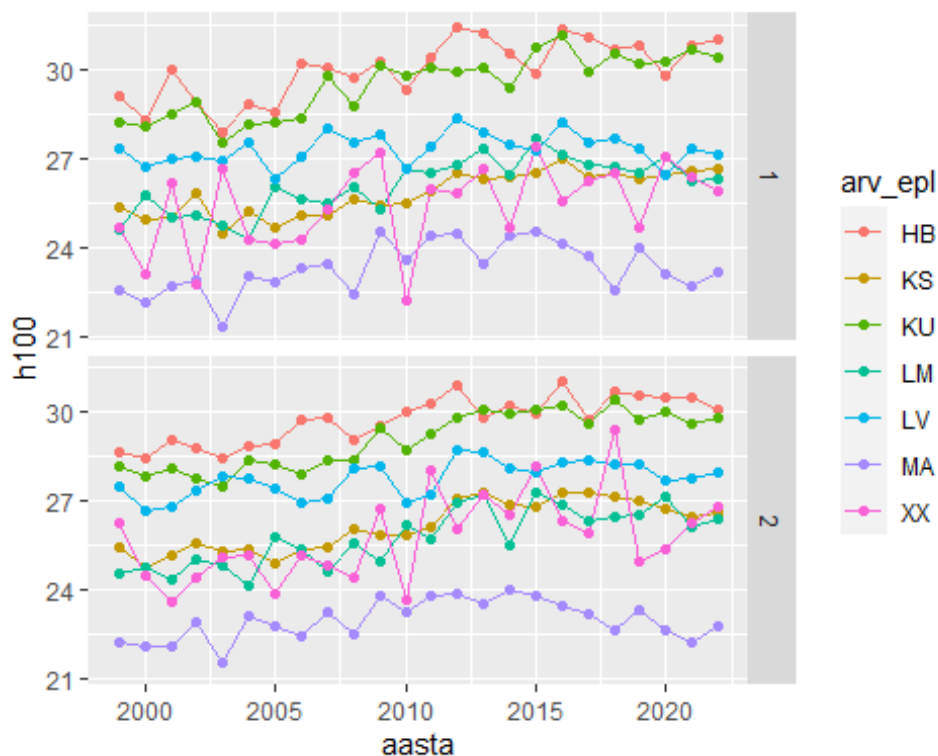
Tulemus: keskmise kõrguse kasv ilmneb kõikidel puuliikidel (v.a. LV) ajaperioodil 2004-2016; alates 2017 keskmise kõrguse kasv peatub või isegi langeb.

4) Edasi uurime H100 käitumist proovitüki liigi järgi (1- tagavara tükk; 2- kasvukohatükk)

```
joonisele <- korgusandmed|>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1,prtliik %in% c(1,2))|>
group_by(aasta,arv_epl,prtliik)|>summarize(h100=mean(arv_h100))

## `summarise()` has grouped output by 'aasta', 'arv_epl'. You can override using
## the `.groups` argument.

ggplot(joonisele,aes(x=aasta,y=h100))+geom_line(aes(group=arv_epl,color=arv_epl))+geom_point(aes(group=arv_epl
,color=arv_epl))+facet_grid(rows=vars(prtliik))
```



Joonis 6

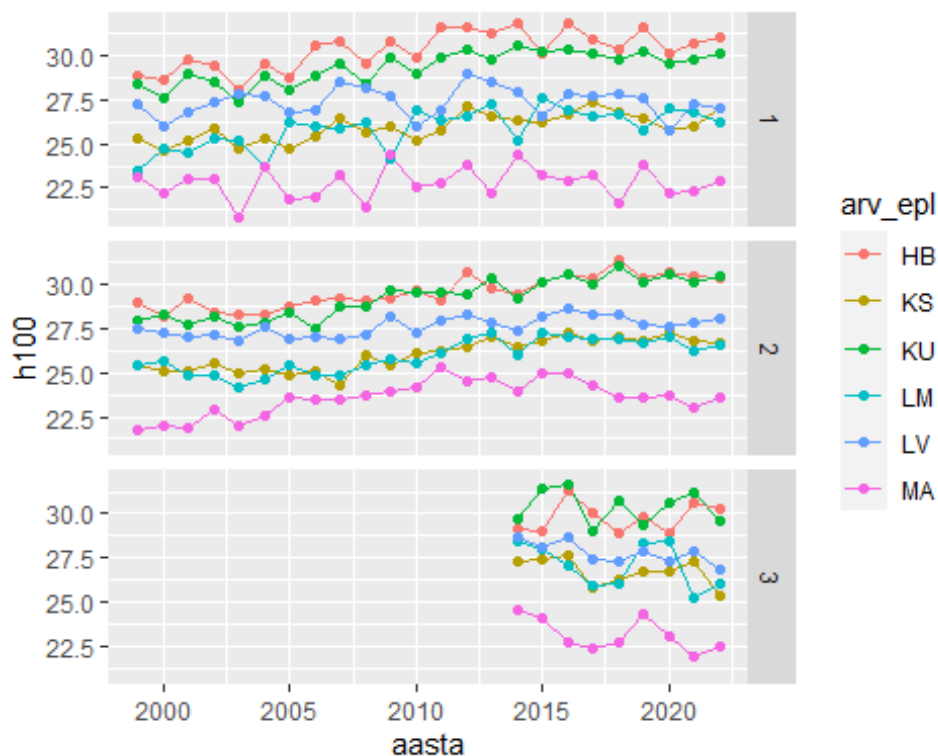
Tulemus: üldine kasvudünaamika on tagavara- ja kasvukohatükkidel sama, seda ka puuliikide lõikes.

5) Vaade traktitüübi järgi (1- alaline; 2- ajutine; 3- alaline uus)

```
joonisele <- trakt|>select(atrakt_id,tr_tyyp)|>inner_join(korgusandmed) |>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1,arv_epl!="XX")>group_by(aasta,arv_epl,tr_tyyp)|>summarize(h100=mean(arv_h100))
```

Joining, by = "atrakt_id"
`summarise()` has grouped output by 'aasta', 'arv_epl'. You can override using
the `groups` argument.

```
ggplot(joonisele,aes(x=aasta,y=h100))+geom_line(aes(group=arv_epl,color=arv_epl))+geom_point(aes(group=arv_epl,color=arv_epl))+facet_grid(rows=vars(tr_tyyp))
```



Joonis 7

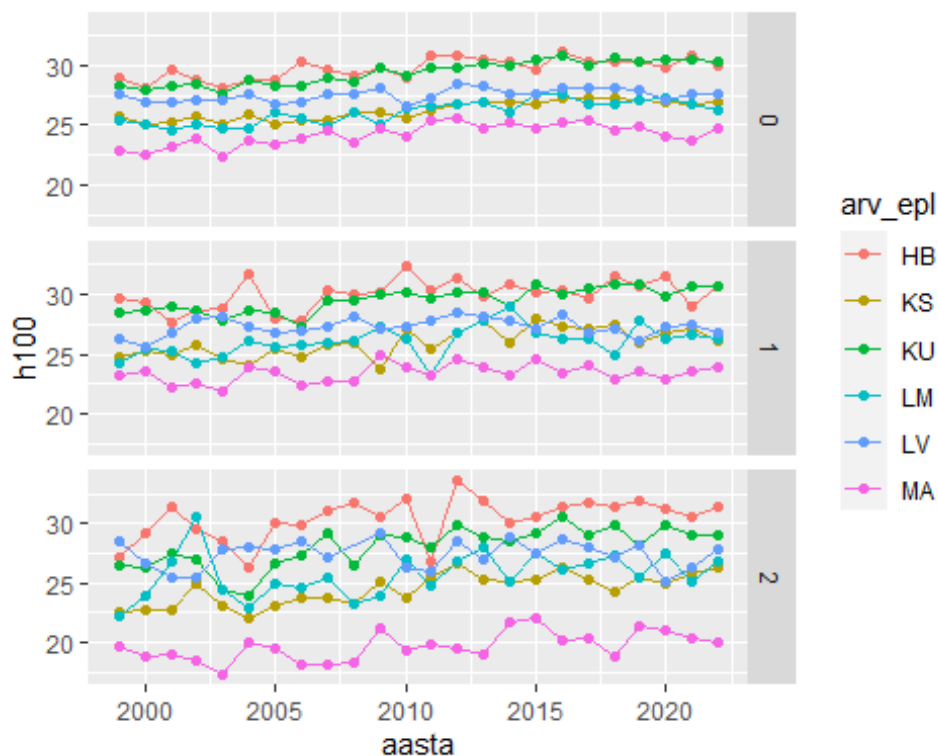
Tulemus: H100 kasv aastail 2004-2016 on ühtviisi nähtav nii alalistel kui ka ajutistel proovitükkidel.

6) Vaade majanduskategooria järgi (0- tulundusmets; 1- kaitsemets; 2- rangekaitse)

```
joonisele <- korgusandmed|>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1,arv_epl!="XX")>
group_by(aasta,arv_epl,majandkategooria)|>summarize(h100=mean(arv_h100))

## `summarise()` has grouped output by 'aasta', 'arv_epl'. You can override using
## the `groups` argument.

ggplot(joonisele,aes(x=aasta,y=h100))+geom_line(aes(group=arv_epl,color=arv_epl))+geom_point(aes(group=arv_epl
,color=arv_epl))+facet_grid(rows=vars(majandkategooria))
```

Joonis 8

Kokkuvõte (H100):

Ülaltoodud analüüs näitas, et H100 muutused ajas on selgelt olemas, kusjuures puude kõrguse kasv on kiirenenud eriti ajaperioodil 2004-2017. Osade puuliikide (KU, KS, LM, HB) korral on kasvuefekt tugevam, teiste (LV, MA) osas muutused on tagasihoidlikumad või puuduvad üldse. Nimetatud tendentsid ilmnevad ühtviisi nii traktitüüpide kui ka proovitüki liikide ja majanduskategooriate lõikes.

4.3. Täiendav metoodiline analüüs H100 muutuste uurimisel

Järgnevalt uurime H100 muutumist detailsemalt, kontrollides seejuures ka üle, kas H100 muutused ei tulene hoopis mõnest metoodiliselt muutusest SMI ajaloo vältel. Näiteks kontrollime 1) kas H100 arvutusmudel on olnud kogu aeg sama, 2) kas muutused ei tulene mitte võimalikest süstemaatilistest nihetest keskmise vanuse ja kõrguse hindamisel, mis on H100 mudeli sisendid (nt mõõtjad vahetuvad!), 3) kas kõrguse muutused ei tulene lihtsalt metsa vanuselise struktuuri muutustest. Viimasele küsimusele vastamiseks tuleks kõrguse muutusi uurida üksikute vanuseklasside lõikes.

4.3.1. H100 mudeli kontroll

Kõigepealt selgitame välja, kas h100 keskmise taseme muutus ei ole seotud mitte mudeli parameetrite muutumisega (st kas arvutusvalemi kordajad on ajas muutunud). Selleks leiame 2023 saadud sql koodis toodud funktsiooni mdl_h100 abil kõikide proovitükiosade h100 väärtused ja võrdleme andmetes olnud väärtustega.

```
mdl_100 <- function(puuliik, vancur, korcur, vanpred){  
  C1 <- if_else(puuliik %in% c('MA', 'TA', 'SA', 'VA', 'JA', 'KP'),0.7283,  
    if_else(puuliik %in% c('KU', 'LH', 'TS', 'NU', 'TO'),0.7977,0.7298))  
  C2 <- if_else(puuliik %in% c('MA', 'TA', 'SA', 'VA', 'JA', 'KP'),-0.0109,  
    if_else(puuliik %in% c('KU', 'LH', 'TS', 'NU', 'TO'),-0.0137,-0.0161))  
  C3 <- if_else(puuliik %in% c('MA', 'TA', 'SA', 'VA', 'JA', 'KP'),1.3925,  
    if_else(puuliik %in% c('KU', 'LH', 'TS', 'NU', 'TO'),1.6116,1.346))  
  IPF1 = korcur*(1+C1*((50/vancur)^C3-1))/(1-C2*korcur*((50/vancur)^C3-1))  
  return(IPF1 / (1+(C1+C2 * IPF1) * ((50 / vanpred)^C3-1)))  
}  
arvutatud <- korgusandmed|>filter(maakategooria %in% c("M"),vanus>15,inv_korgus>=1,arv_epl!="XX")|>  
  mutate(uus_h100=mdl_100(peapuuliik,vanus,inv_korgus/10,100))  
  
sum(abs(arvutatud$arv_h100-arvutatud$uus_h100)>=0.1)  
  
## [1] 0
```

Tulemus: Seega kogu aeg on kasutatud sama H100 valemit ning järelikult H100 muutused tulenevad mudeli sisendite (kõrgus, vanus) jaotuse muutustest.

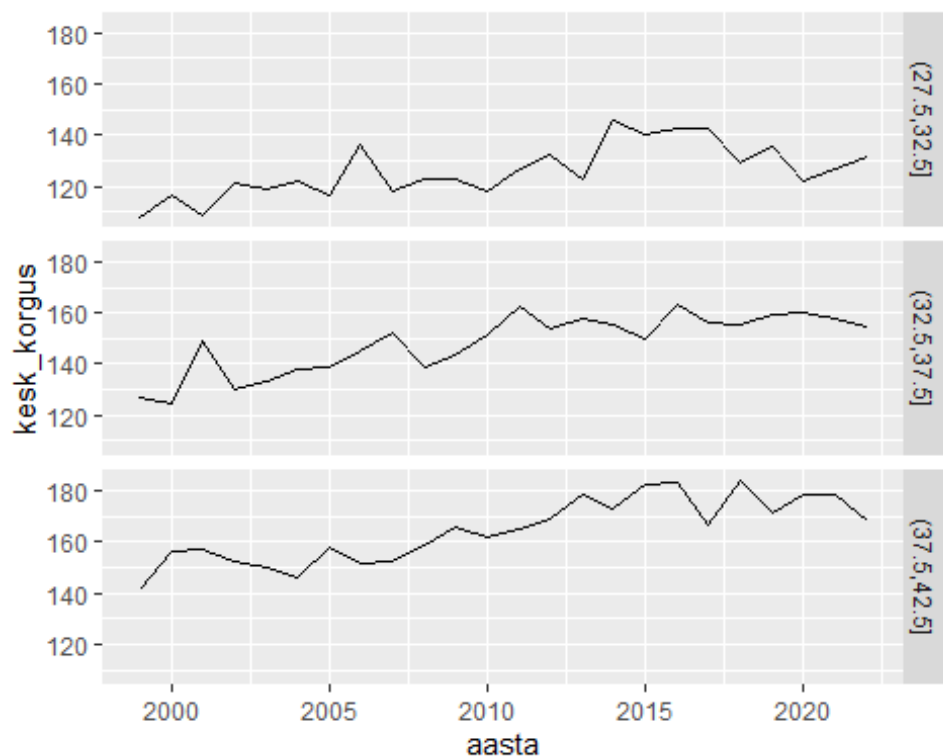
4.3.2. Kõrguse kasv vanuserühmade kaupa

Vaatame nüüd, kas kõrgus on **sama vanuse** korral kasvanud. Selleks jaotame kõrguse 5-aastastesse rühmadesse ja leiame aastate kaupa keskmise kõrguse vanuserühmades.

```
korgus_vanus <- arvutatud |> mutate(vanuskl=cut(vanus,breaks=c(seq(12.5,162.5,by=5),Inf)))|>  
  group_by(aasta,peapuuliik,vanuskl)|>summarize(kesk_korgus=mean(inv_korgus))|>ungroup()  
  
## `summarise()` has grouped output by 'aasta', 'peapuuliik'. You can override  
## using the `.groups` argument.
```

Nüüd on võimalik uurida konkreetsete puuliikide keskmise kõrguse muutusi ajas vanuserühmade kaupa. Kuuse puhul on paari vanuserühma jaoks graafikud allpool.

```
korgus_vanus|>filter(peapuuliik=="KU",as.numeric(vanuskl) %in% 4:6) |>ggplot(aes(x=aasta,y=kesk_korgus))+geom_  
line()+facet_grid(rows=vars(vanuskl))
```

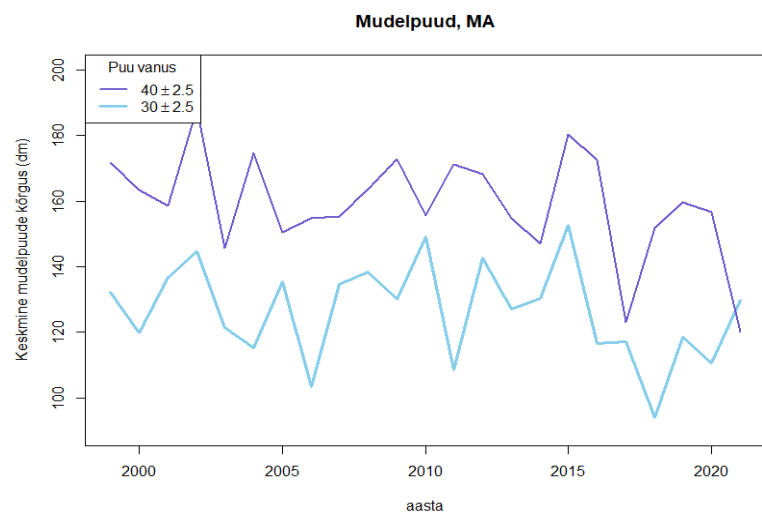
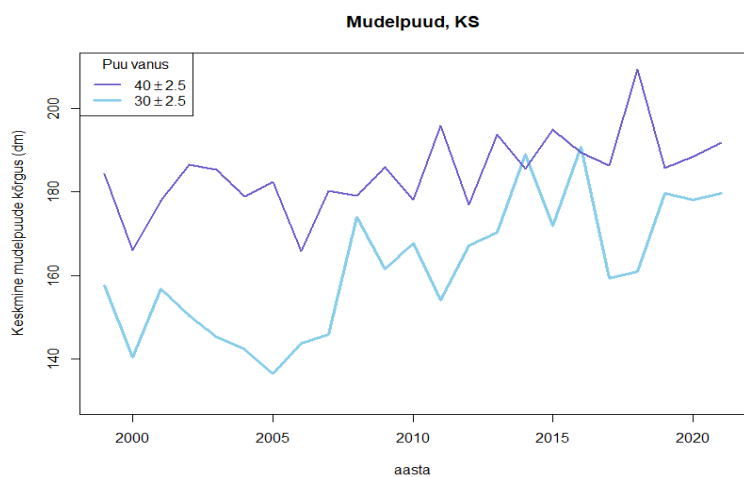
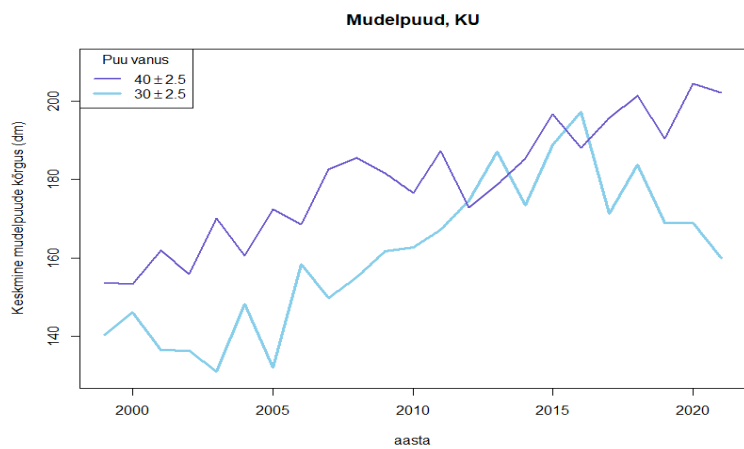


Joonis 9

Tulemus: Viimaselt graafikult on näha, et 23 aastaga on kõigis kolmes vanuseklassis kuuskede keskmine kõrgus tõusnud ca 3m.

4.3.3. H100 muutused mudelpuudel

Järgnevas kontrollime veel lisaks, et kas muutused keskmises kõrguses ei tulene mitte tunnuste mõõtmismetoodika muutusest ajas (proovitüki peapuuliigi keskmist vanust ja keskmist kõrgust hinnatakse silma järgi, samas mõõtjad aeg-ajalt vahetuvad) või on see efekt täheldatav ka mudelpuude korral, kus mõõtmised on objektiivsed. Mudelpuude vanus ja kõrgus on mõõdetud hea täpsusega, kuna tegu on instrumentaalsete mõõtmistega.



Joonis 10

Tulemus (mudelpuud): viimased 3 joonist näitavad ka mudelpuude keskmise kõrguse kasvu, seda eriti kuuse ja kase osas. Mändide korral mudelpuude keskmise kõrguse tõusu

pole märgata, viimase kuue aasta jooksul ilmneb pigem kõrguse langus. Teised puuliigid on mudelpuude hulgas esindatud liiga vähe, et taolisi ajalisi muutusi jälgida.

4.3.4. Kokkuvõtte H100 ja kõrguse dünaamika analüüsist

Selgus, et nii kõikide puude kokkuvõttes kui ka üksikute vanuseklasside piires puude keskmised kõrgused on viimase paarikümne aasta jooksul selgelt tõusnud, kusjuures tõus on olnud eriti kiire SMI II ja III perioodi jooksul (2004 -2016), pärast mida tõus on vaibunud. Kasvu kiirenemine kehtib eriti kuuskede ja kaskede kohta, kus 30-40 aastased puud on nüüd keskmiselt 2-3 meetrit kõrgemad kui varem, samas mändide korral kasvu kiirenemise tendentsi ei ilmne.

4.4. Optimaalse ajaakna leidmine kõrguse mudeli jaoks

4.4.1. Sissejuhatus

Nagu eelnev analüüs näitas, on teatud puuliikide kasv viimasel paarikümnel aastal kiirenenud. Seda tendentsi tuleks arvestada ka metsanduslike mudelite koostamisel ja kasutamisel. Seetõttu tekib küsimus, kui pikka aegrida tuleks kõrguse ja hektaritagavara mudelite kalibreerimisel kasutada, et ühelt poolt kaasas käia looduslike tingimuste sh kliima muutustega (mudeli süstemaatilise vea vähendamise eesmärgil), aga teiselt poolt arvestada, et liiga lühike andmeperiood suurendab mudeli vea juhuslikku komponenti.

4.4.2. Erineva pikkusega ajaakende võrdlus kõrgusmudeli korral

Järgneva analüüsiga püüame välja selgitada, millise pikkusega ajaaken (andmeintervall) on optimaalne kõrgusmudeli

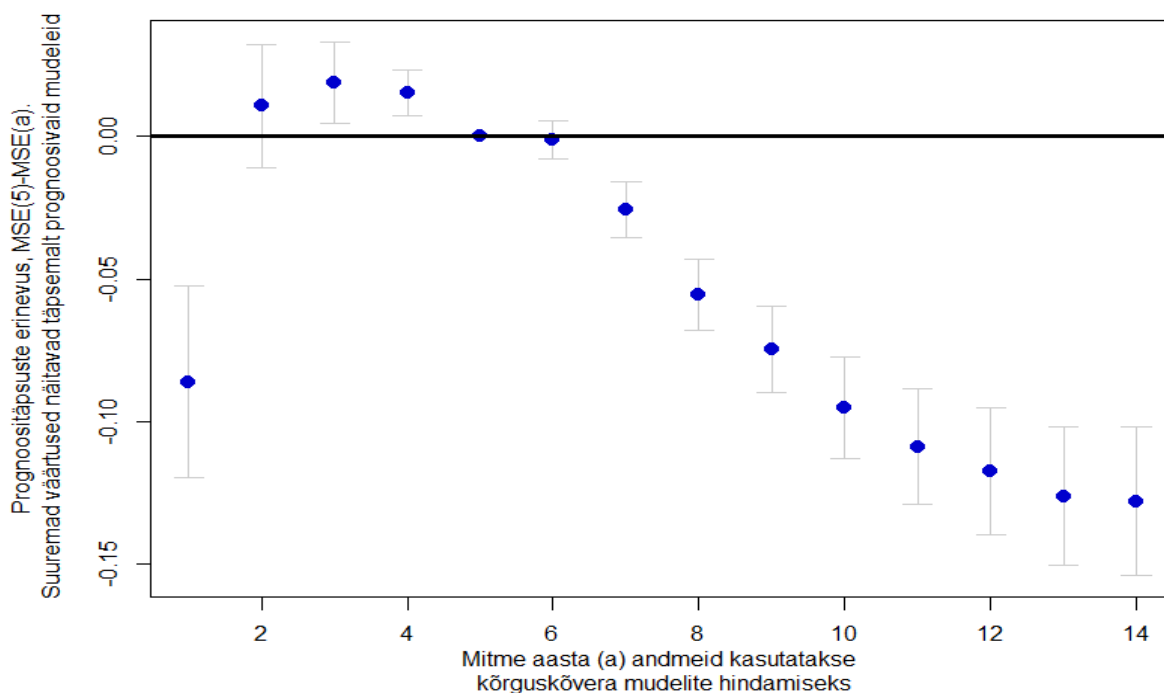
$$h = 1,3 + (a_1 + a_2 \cdot H100 + a_3 \cdot OnSaar) \left(\frac{d_{1,3}}{d_{1,3} + b_1} \right)^{c_1 + c_2 \cdot H100}$$

rekalibreerimiseks, et oleks tagatud mudeli keskmiselt suurim prognoositäpsus RMSE mõttes (RMSE võtab kokku nii süstemaatilise kui ka juhusliku vea).

Võrdlusbaasiks valisime 5-aastase ajaintervalli, mis mõningatel kaalutlustel tundus olevat küllaltki mõistlik kandidaat otsitavale optimaalsele ajaaknale ja millega sai võrreldud kõiki teisi ajaintervalle. Järgnevas on läbi viidud lihtne analüüs: iga üksiku puu kõrgust on prognoositud 5 viimase aasta põhjal loodud mudeli abil ja võrdluseks ka a eelneva aasta andmete põhjal loodud mudeli abil. Leiame mõlemal juhul tekkiva ruutvea ja arvutame nende vahe. Negatiivsed väärtused näitavad, et täpsemaid prognoose saaksime viimase 5 aasta andmete põhjal hinnatud mudeli abil, positiivsed väärtused vihjavad, et parem oleks kasutada a aasta põhjal hinnatud mudelit.

Puu kõrguse prognoosimisel tehtavat vigat on arvatud ristvalideerimist kasutades – prognoositava puu enda andmeid pole kummagi mudeli hindamisel kasutatud.

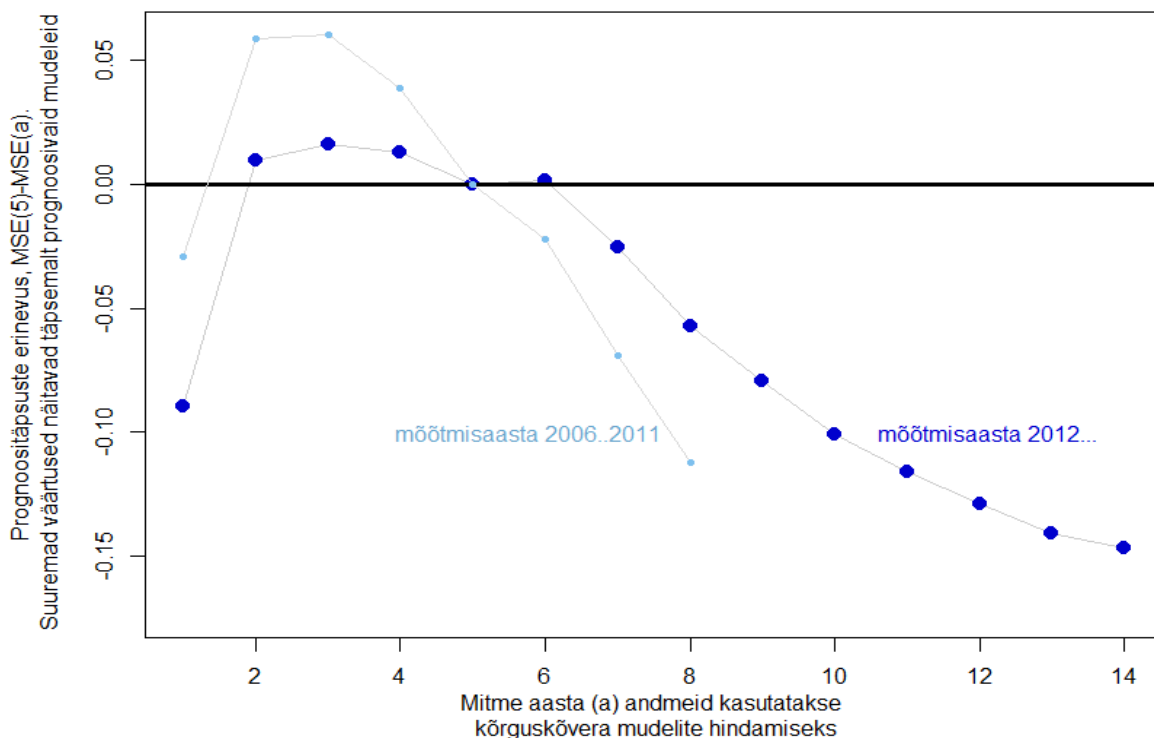
Allolev joonis on saadud ainult 2012 ja hiljem mõõdetud puud kasutades – need on puud, mille puhul on olemas vähemalt 14 aastane mõõteajalugu (esimesed mõõtmised tehti 1999). Selliste puude kõrgust saab prognoosida 14 erinevat ajaakent kasutava mudeliga ($a = 1, 2, 3, \dots, 14$).



Joonis 11

Jooniselt 11 näeme, et kõige väiksem prognoosiviga esineb juhul, kui prognoosimudel on hinnatud 3 või 4 eelneva aasta põhjal, järgnevad 5 ja 6-aastased ajaaknad. Praktilisest seisukohast lähtudes on nähtavasti otstarbekaim ajaakna valik siiski 5 või 6 aastat, kuna need haaravad kogu 5-aastase SMI tsükli andmed.

Kuna eelmine graafik puudutas ainult puud, mis olid mõõdetud 2012 või hiljem, siis järgmisel graafikul on juurde lisatud üks joon (helesinine õrn joon), mis on saadud aastail 2006-2011 mõõdetud puude korral – nende puude korral on võimalik kasutada ainult kuni 8 aastat ajalugu mudelite hindamiseks. Näeme, et ka siin mõistlikeim ajaaken 5 või 6 aastat.

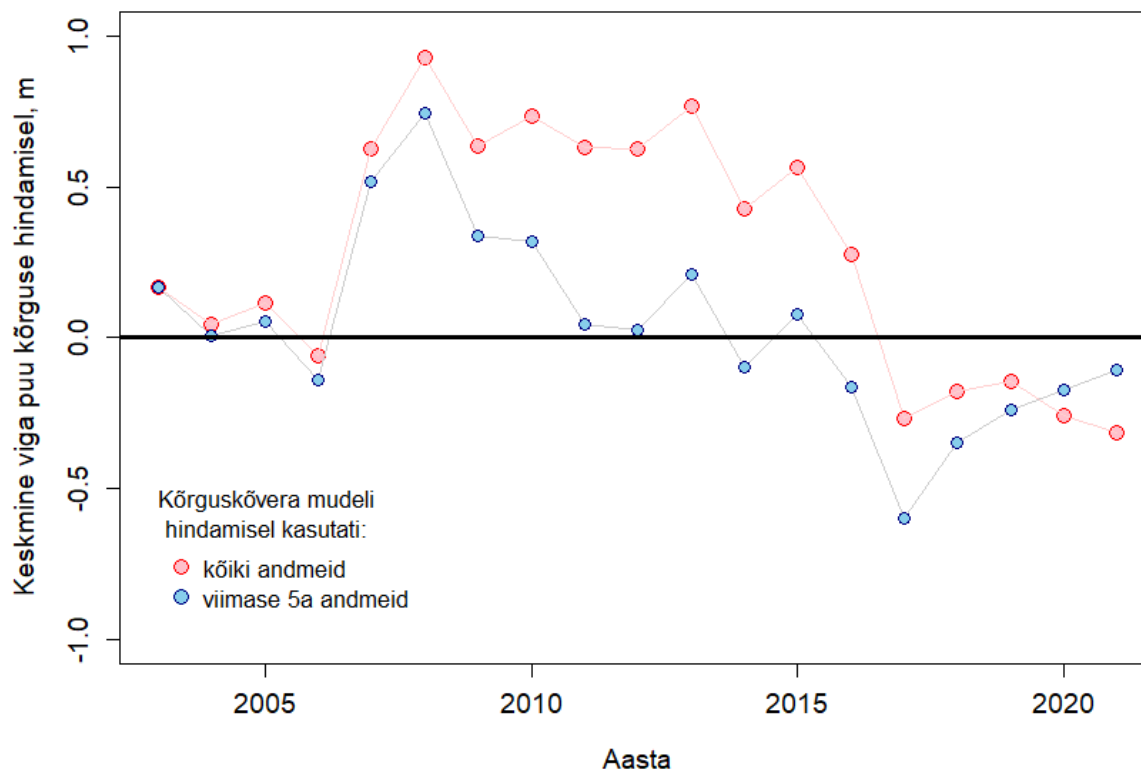


Joonis 12

4.4.3. Viieaastase ajaakna võrdlus senise mõõteajaloo

Eelolevatelt graafikutelt 11 ja 12 nägime, et 5- või 6-aastane ajaaken on parim variant kõrguskõvera kalibreerimiseks. Urime nüüd lähemalt, kui hästi näiteks 5-aastase ajaaknaga kõrguse mudel töötab võrreldes mudeliga, mis kasutab igal hetkel kogu senist mõõteajalugu. Kas 5-aastast ajaakent kasutatav mudel on **alati** parem kui kogu senist mõõteajalugu kasutatav mudel või on ta parem **enamikel juhtudel**?

Esimesel joonisel on mudeleid võrreldud nende kasutamisel tekkiva **süsteemaatilise vea** seisukohast. Täpsemalt, kõrguskõvera mudeli parameetreid on hinnatud kahel viisil – kasutades kõiki tolleks hetkeks saadaolevaid mõõtmistulemusi (punased punktid) või kasutades kõigest viimase viie aasta vaatlusandmeid (sinised punktid). Punktid graafikul näitavad, kui suur on mingil aastal keskmine viga puude kõrguste hindamisel (diameetri järgi kõrguse hindamine, kasutades kõrguskõvera mudelit) mingil konkreetsel aastal. Kui punkt näitab näiteks väärtust 0,5, siis hinnati kõrguskõvera mudelit kasutades sellel aastal mõõdetud puude kõrguseid keskmiselt 0,5m võrra väiksemaks, kui nad tegelikult olid. Mida lähemal on mingi punkt paksule tumedale nulljoonele, seda väiksem on antud aastal mõõdetud puudele tehtud kõrgusproгноosi süsteemaatiline viga. Kuna sinised punktid on enamasti nulljoonele lähemal, siis võib öelda, et kõrgusmudeli rek calibreerimine viimase viie aasta andmete põhjal annab valdavalt (aga rangelt võttes mitte iga kord) täpsemad prognoosid kui kogu senist mõõteajalugu kasutatav mudel.

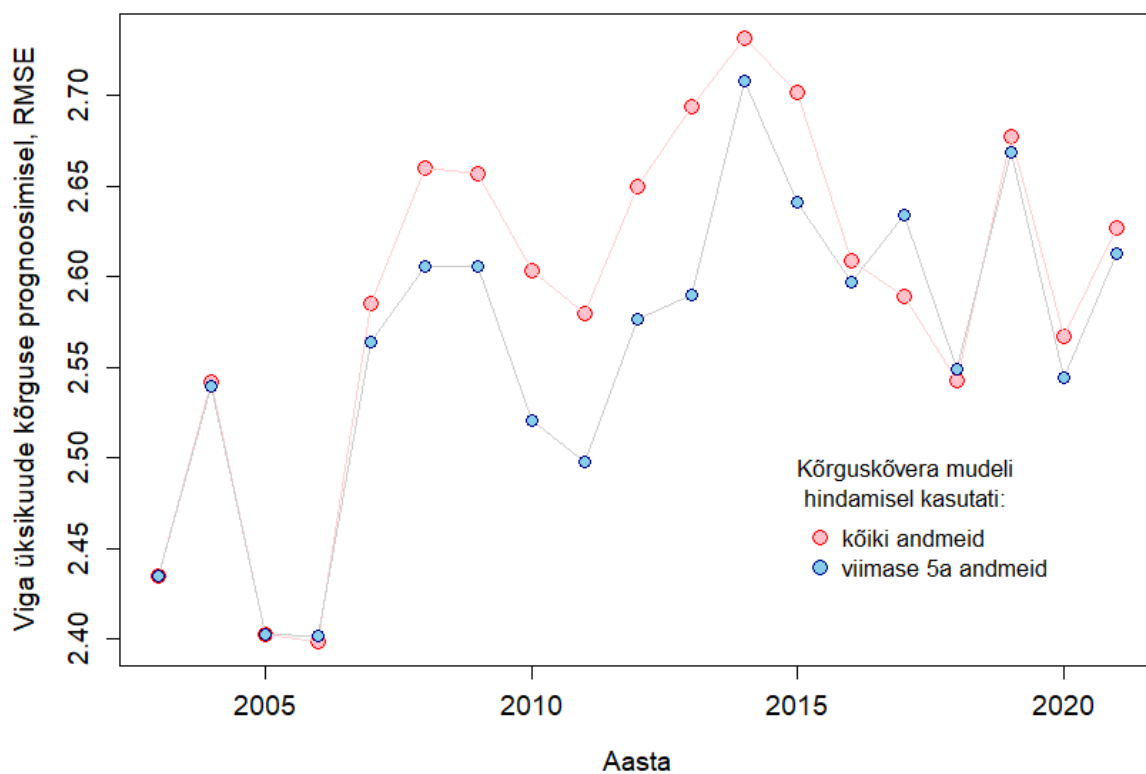


Joonis 13

Joonise 13 täpsem saamislugu on järgmine:

Eemaldame andmestikust 5% traktidest – täpsemalt nendel traktidel tehtud mudelpuude mõõtmised. Hindame allesjäänud andmete põhjal kaks kõrguskõvera mudelit – ühel juhul kasutades kõiki selleks hetkeks olemasolevaid vaatluseid (nn kõiki andmeid kasutav mudel) ja teise mudeli kasutades kõigest viimase 5 aasta vaatlustulemusi. Mõlemat mudelit kasutades prognoosime mudeli hindamisel mittekasutatud traktidel mõõdetud mudelpuude kõrguseid. Kogu protseduuri kordame 20 korda – eemaldame järgmised 5% traktidest ja prognoosime nendel traktidel tehtud mõõtmistulemusi ülejäänud 95% traktidel mõõdetud mudelpuude järgi jne. Lõpuks leiame tekkinud prognoosivigade keskmised – kui hinnatud mudel kirjeldab adekvaatselt antud aastal tehtud vaatluseid, siis peaks antud aastal tehtud vaatluste jaoks keskmine prognoosiviga tulema 0. Kui konkreetsel aastal on prognoosivigade keskmine märkimisväärselt nullist erinev, viitab see olukorrale, kus mudel ei kirjelda adekvaatselt vasta antud aastal looduses toimuvat.

Järgmisel joonisel on võrreldud 5-aastast ajaakent kasutava kõrguse mudelit ja kogu mõõteajalugu kasutavat mudelit **juhuslike prognoosivigade** seisukohast (RMSE). Näeme taas, et valdaval enamikul juhtudest 5-aastase ajaaknaga mudel annab väiksema vea kui kogu mõõteajalugu kasutav mudel.



Joonis 14

4.4.4. Kokkuvõte

Kokkuvõte: Kõrgusmudeli kalibreerimiseks sobivaimad ajaaknad on 5 ja 6 aastat - sellise andmeintervalli abil kalibreeritud kõrgusmudelid on täpseimad RMSE mõttes (hõlmab nii süstemaatilist kui ka juhuslikku prognoosiviga). Detailsem analüüs näitas, et valdavalt enamikul juhtudel 5-aastast ajaakent kasutatav mudel annab tõepoolest väiksema vea kui kogu mõõteajalugu kasutatav mudel.

Versioon 2.0
Kalev Pärna 31.05.2024



4.4.5. Programmi kood (joonised 11 ja 12)

```
# Teek algandmete sisselugemiseks
library("RSQLite")
# Teek mudelite numbriliseks hindamiseks
library("gslns")

# Vajalike algandmete lugemine R-i

## connect to db
con <- dbConnect(drv=RSQLite::SQLite(), dbname="c:/konsult/SMI2/smi_koik_ilmaXY.sqlite")
## list all tables
tables <- dbListTables(con)
## exclude sqlite_sequence (contains table information)
tables <- tables[tables != "sqlite_sequence"]
## create a data.frame for each table
for (i in seq(along=tables)) {
  assign(tables[i], dbGetQuery(conn=con, statement=paste("SELECT * FROM '", tables[[i]], "'", sep="")))
}
ajut=dbGetQuery(conn=con, statement="
  select k.*, t.aasta, t.tr_tyyp, maakategooria, inv_korgus, arv_h100, vanus, peapuulik, psees,
  on_sees, o.onsaar
  from trakt t, prt p, prtosa o, tpuulik x, mudelpuud k
  where t.atrakt_id = p.atrakt_id and p.prt_aid = o.prt_aid and p.aproovitykk_id =
  k.aproovitykk_id and
  k.osa_nr = o.osa_nr and k.puulik = x.kood and x.puu = 2 and kahj_aste is null")

ind=(ajut$maakategooria %in% c("M","MM") &
!(is.na(ajut$vanus)|is.na(ajut$d_13)|is.na(ajut$h_puu)) & ajut$vanus>15 & ajut$d_13>50)
valim1=ajut[ind,]

# Puude grupid. Iga grupi jaoks hinnatakse eraldi mudel.
# Ametlik variant (ei sisalda "põõsaid")
plgrp <- list(
```

Version 2.0
Kalev Pärna 31.05.2024



```
"MA" = c("MA", "LH", "SD"),  
"KU" = c('KU', 'NU', 'TO', 'TS'),  
"KS" = c('KS', 'PN', 'TL', 'EL'),  
"HB" = c('HB', 'PP'),  
"LM" = c("LM"),  
"LV" = c("LV"),  
"XX" = c("TA", "SA", "VA", "JA", "KP")  
)
```

```
puuliike=length(plgrp)
```

```
# jagame kõik traktid 20-sse gruppi. Hiljem jätame ühes grupis (5% traktidest) mõõdetud puud mudeli hindamisest välja
```

```
# hindame mudeli(d) vaid 95% andmete pealt ja vaatame, kui hästi nad väljajäänud osa prognoosivad:
```

```
n_traktigruppe=20
```

```
ntrakte = length(table(valim1$atrakt_id))
```

```
set.seed(1)
```

```
grupid = sample(1:n_traktigruppe, ntrakte, replace=TRUE)
```

```
traktinimed=names(table(valim1$atrakt_id))
```

```
ajut=rep(NA, dim(valim1)[1])
```

```
for (i in 1:ntrakte){
```

```
  ajut[valim1$atrakt_id == traktinimed[i] ] = grupid[i]
```

```
}
```

```
valim1$traktigrupp=ajut
```

```
maxaastaid=length(table(valim1$aasta))
```

```
maxaastaid
```

```
# Siia salvestame mudelite abil prognoositud puu kõrgused (NB! mudeli hindamisel pole prognoositava puu enda andmeid kasutatud):
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# Esimene tulp - antud tüki prognoos sama aasta andmete pealt; teine tulp - prognoos käesoleva aasta ja eelmise aasta põhjal jne.
```

```
prognoos_maatriks=matrix(NA, nrow=dim(valim1)[1], ncol=maxaastaid)
```

```
koondumine = matrix(NA, nrow=dim(valim1)[1], ncol=maxaastaid)
```

```
# Eemaldame ühe grupi valimist, seejärel hindame mudeli kasutades valitud arvu aastate mõõtmistulemusi
```

```
for ( eemaldagrupp in 1:n_traktigruppe){  
  print (paste("Traktigrupp:", eemaldagrupp, "/", n_traktigruppe ))
```

```
  valim2=valim1[valim1$traktigrupp!=eemaldagrupp,]
```

```
# millise aasta andmeid üritame prognoosida?
```

```
for ( prognoositav_aasta in (1999):(1999+maxaastaid-1) ){
```

```
  print (paste("Prognoosin aastat:", prognoositav_aasta ))
```

```
  flush.console()
```

```
# Mitme eelneva aasta põhjal hindame kõrguskõvera mudelid:
```

```
for (aastaid_mudelis in 1:(prognoositav_aasta-1999+1)){
```

```
# Siia salvestame prognoosivad mudelid - iga puuliigi(grupi) jaoks üks mudel
```

```
mudelid=list(puuliike)
```

```
# Siia läheb kirja informatsioon selle kohta, kas hinnatud mudel koondus
```

```
koondumine_abi=rep(NA, puuliike)
```

```
# Valime puuliigi
```

```
for (xpl in 1:puuliike){
```

```
# Mudeli loomiseks kasutatav ajavahemik:
```

```
mudeli_ajavahemik=(prognoositav_aasta-aastaid_mudelis+1):prognoositav_aasta
```

```
nls.tul=NULL
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# Nopime mudeli hindamiseks välja andmestiku - vaid sobivast liigist puud (puuliik) ja takseerimisaasta kuulub soovitud vahemikku
```

```
valim <- valim2[ $\text{valim2}\$$ puuliik  $\%in\%$  plgrp[[xpl]] &  $\text{valim2}\$$ aasta  $\%in\%$  mudeli_ajavaheikk,]
```

```
# Hindame mudeli kasutades moodustatud andmestikku valim:
```

```
# -----
```

```
kkpars <- function(b1, c1, c2, h, d, h100, onsaar)
```

```
{
```

```
  x <- d/(d+b1)
```

```
  y <- x^(c1+h100*c2)
```

```
  w <- (h-1.3)/y
```

```
  lm.t <- lm(w~h100+onsaar)
```

```
  cf <-<- coef(lm.t)
```

```
  1.3+(cf[1]+cf[2]*h100+cf[3]*onsaar)*(d/(d+b1))^(c1+h100*c2)
```

```
}
```

```
# Kõrguskõvera parameetrite arvutamine kahe-astmelisena
```

```
cf <- NULL
```

```
nls.tul <- try(gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim,  
start=list(b1=4, c1=2, c2=0.1),
```

```
  algorithm = "Imacel"), silent=TRUE)
```

```
# gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim, start=list(b1=4,  
c1=2, c2=0.1), trace=TRUE)
```

```
if(inherits(nls.tul, "try-error")) {print("Probleemid mudeli koondumisega ...")} else {
```

```
  pars <- as.list(nls.tul$m$getPars())
```

```
  pars$a1 = cf[[1]]
```

```
  pars$a2 = cf[[2]]
```

```
  pars$a3 = cf[[3]]
```

```
nls.tul.2 <- gsl_nls( h_puu/10  
~1.3+(a1+a2*arv_h100+a3*onsaar)*((d_13/10)/((d_13/10)+b1))^(c1+arv_h100*c2),
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
        valim, start=pars, algorithm="lmaccel")
    }

    mudelid[[xpl]] = nls.tul.2

} # lõppeb tsükkel üle puuliikide

# Leiame prognoosid mudeli hindamiseks mittekasutatud puudele:

for (i in 1:puuliike){
  indeks=valim1$puuliik %in% plgrp[[i]] & valim1$aasta==prognoositav_aasta &
  valim1$traktigrupp==eemaldagrupp
  valim <- valim1[indeks,]
  prognoos_maatriks[indeks,aastaid_mudelis]=predict(mudelid[[i]], valim)
  koondumine[indeks, aastaid_mudelis] = koondumine_abi[[i]]
}

} # lõppeb tsükkel: aastaid mudelis
} # lõppeb tsükkel: prognoositav_aasta
} # lõppeb tsükkel: traktigrupp

#-----

# Leiame prognoosivead (tegelik arvatud klupitud puude järgi - mudeli prognoos):

veamaatriks=matrix(NA, nrow=dim(prognoos_maatriks)[1], ncol=dim(prognoos_maatriks)[2])
for (i in 1:maxaastaid){
  veamaatriks[,i]=valim1$h_puu/10-prognoos_maatriks[,i]
}

kaugeajalugu=14
ind=(valim1$aasta-1999+1)>=kaugeajalugu

hinnang=rep(NA, kaugeajalugu)
```

Versioon 2.0
Kalev Pärna 31.05.2024



Ualumine=rep(NA, kaugeajalugu)

Ulylemine=rep(NA, kaugeajalugu)

```
for (i in 1:kaugeajalugu){  
  testitulemus=t.test(veamaatriks[ind,5]**2-veamaatriks[ind,i]**2)  
  hinnang[i]=testitulemus$estimate  
  Ualumine[i]=testitulemus$conf.int[1]  
  Ulylemine[i]=testitulemus$conf.int[2]  
}
```

Tulemuste visualiseerimine

Joonis 11

Kõrguskõverate mudelite suhteline täpsus võrreldes 5 aasta andmeid kasutavate mudelitega, koos usalduspiiridega.

```
windows(width=4*3, height=3*3)
```

```
par(mar=c(5.5,5.5,3.1,2.1))
```

```
plot(1:kaugeajalugu, hinnang, ylim=range(c(Ualumine, Ulylemine), na.rm=T), xlab="Mitme aasta (a)  
andmeid kasutatakse \n kõrguskõvera mudelite hindamiseks", ylab="Prognoositäpsuste erinevus,  
MSE(5)-MSE(a).\n Suuremad väärtused näitavad täpsemalt prognoosivaid mudeleid")
```

```
arrows(1:kaugeajalugu, Ualumine, 1:kaugeajalugu, Ulylemine, angle=90, length=0.1, code=3,  
col="gray80")
```

```
points(1:kaugeajalugu, hinnang, pch=20, col="blue3", cex=2)
```

```
abline(h=0, lwd=3)
```

Joonis 12 - Kui pikka ajalugu kasutada, 2011a andmete ja varasemate aastate jaoks vs 2012 ja hilisemad aastad

```
ind2= valim1$aasta %in% 2006:2011
```

```
hinnang2=rep(NA, 8)
```

```
Ualumine2=rep(NA, 8)
```

```
Ulylemine2=rep(NA, 8)
```

```
for (i in 1:8){
```

```
  testitulemus2=t.test(veamaatriks[ind2,5]**2-veamaatriks[ind2,i]**2)
```

```
  hinnang2[i]=testitulemus2$estimate
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
Ualumine2[i]=testitulemus2$conf.int[1]
Ulylemine2[i]=testitulemus2$conf.int[2]
}

windows(width=4*3, height=3*3)
par(mar=c(5.5,5.5,3.1,2.1))
plot(1:kaugeajalugu, hinnang, ylim=range(c(Ualumine, Ulylemine, hinnang2), na.rm=T),
      xlab="Mitme aasta (a) andmeid kasutatakse \n kõrguskõvera mudelite hindamiseks",
      ylab="Prognoositäpsuste erinevus, MSE(5)-MSE(a).\n Suuremad väärtused näitavad täpsemalt
      prognoosivaid mudeleid",
      type="l", col="gray80")
# arrows(1:kaugeajalugu, Ualumine, 1:kaugeajalugu, Ulylemine, angle=90, length=0.1, code=3,
# col="gray80")
# arrows((1:8)+0.05, Ualumine2, (1:8)+0.05, Ulylemine2, angle=90, length=0.1, code=3,
# col="lightblue")

points(1:kaugeajalugu, hinnang, pch=20, col="blue3", cex=2)
abline(h=0, lwd=3)

lines(1:8, hinnang2, col="gray85")
points(1:8, hinnang2, pch=20, col="skyblue2", cex=1)
text(10.6, -0.1, "mõõtmisaasta 2012...", adj=0, col="blue3")
text(7.6, -0.1, "mõõtmisaasta 2006..2011", adj=1, col="skyblue3")
```


Versioon 2.0
Kalev Pärna 31.05.2024



4.4.6. Programmi kood (joonised 13 ja 14)

```
# Algandmete sisselugemine
# -----

library("RSQLite")

## connect to db
con <- dbConnect(drv=RSQLite::SQLite(), dbname="c:/konsult/SMI2/smi_koik_ilmaXY.sqlite")
## list all tables
tables <- dbListTables(con)
## exclude sqlite_sequence (contains table information)
tables <- tables[tables != "sqlite_sequence"]
## create a data.frame for each table
for (i in seq(along=tables)) {
  assign(tables[i], dbGetQuery(conn=con, statement=paste("SELECT * FROM ", tables[[i]], "",
  sep="")))
}

ajut=dbGetQuery(conn=con, statement="
  select k.*, t.aasta, t.tr_tyyp, maakategooria, inv_korgus, arv_h100, vanus, peapuuliik, psees,
on_sees, o.onsaar
  from trakt t, prt p, prtosa o, tpuuliik x, mudelpuud k
  where t.atrakt_id = p.atrakt_id and p.prt_aid = o.prt_aid and p.aproovitykk_id =
k.aproovitykk_id and
      k.osa_nr = o.osa_nr and k.puuliik = x.kood and x.puu = 2 and kahj_aste is null")
dim(ajut)

# Mudelpuud, algselt 72105 kirjet
# dim(mudelpuud)
# [1] 72105 28

# Kaod:

# Eemaldame kahjustusega puud (kahjustuse aste märgitud):
# table(is.na(mudelpuud$kahj_aste))
# 72105 -9543 = 62562

# Eemaldame 401 kirjet mis pole puud (OP, PI, RE, SP, TM):
# table(mudelpuud$puuliik[is.na(mudelpuud$kahj_aste)], exclude=NULL)
# 3+5+1+19+303+2+71
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# 72105 -9543 - 404 = 62161
```

```
# 21 kirje puhul mudelpuu seos prtosa andmestikuga vigane (sellist prt_aid/osa_nr kombinatsiooni ei eksisteeri)
```

```
# Probleemsete mudelpuude id-d, mudelpuud$amudelpuud_id väärtused:
```

```
#      ajut2=merge(mudelpuud, prtosa, by=c("osa_nr", "prto_aid"), all.x=TRUE)
#      ind=is.na(ajut2$aproovitykk_id.y)
#      ajut2$amudelpuud_id[ind]
#      [1] 70906 70905 72244 75083 75736 75888 70684 70683 74799 74205 74712 73511
76309 76308 74099 74183 70682 70681 74396 69328 69327
```

```
#      print(mudelpuud[mudelpuud$amudelpuud_id %in% ajut2$amudelpuud_id[ind],c(3,4,5,
7,8,14,22) ], row.names=F)
#      print(prtosa[prtosa$aproovitykk_id %in% ajut2$aproovitykk_id.x[ind],c(3,9,10)],
row.names=F)
```

```
# Neist proovitükiosata mudelpuudest 17 olid sellised, millel olid terved (kahjustusteta) puud:
# ind2= (mudelpuud$amudelpuud_id %in% ajut2$amudelpuud_id[ind])
# sum(ind2 & is.na(mudelpuud$kahj_aste) & !is.na(mudelpuud$puuliik) &
!(mudelpuud$puuliik%in%c("OP","PI","RE","SP","TM", "")))
# mudelpuud[ind2 & is.na(mudelpuud$kahj_aste) & !is.na(mudelpuud$puuliik) &
!(mudelpuud$puuliik%in%c("OP","PI","RE","SP","TM", ""))],
```

```
# 72105 -9543 -8- 396 -17 = 62141
```

```
# dim(ajut)
```

```
# Mis läheb täiendavalt välja?
```

```
# Kas puu on mõõdetud metsamaal (1 651 mujal / 60 490 metsamaal)
```

```
# table( (ajut$maakategooria %in% c("M","MM")))
```

```
# Kas puu Mõõtmisandmed on olemas (ühel puul puuduvad)?
```

```
# ind=(ajut$maakategooria %in% c("M","MM"))
```

```
# sum(is.na(ajut$vanus[ind])|is.na(ajut$d_13[ind])|is.na(ajut$h_puu[ind]))
```

```
# ajut[ind&(is.na(ajut$vanus)|is.na(ajut$d_13)|is.na(ajut$h_puu)),]
```

```
# Kui palju puid on sellised, mis rahuldavad valikukriteeriumi
```

```
# metsa keskmine vanus>15 ja puu diameeter>5cm (5cm või 5mm?)
```

```
# Selliseid puid on 53255 tükki:
```

```
#      ind=(ajut$maakategooria %in% c("M","MM")) &
!(is.na(ajut$vanus)|is.na(ajut$d_13)|is.na(ajut$h_puu)) )
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# table( ajut$vanus[ind]>15 & ajut$d_13[ind]>50)
# FALSE TRUE
# 7234 53255
```

```
# Korjame välja puud, mis sobivad kõrguskövera hindamiseks:
```

```
ind=(ajut$maakategooria %in% c("M","MM") &
!(is.na(ajut$vanus)|is.na(ajut$d_13)|is.na(ajut$h_puu)) & ajut$vanus>15 & ajut$d_13>50)
```

```
valim1=ajut[ind,]
dim(valim1)
```

```
# Puude grupid. Iga grupi jaoks hinnatakse eraldi mudel.
```

```
# Ametlik variant (ei sisalda "põõsaid")
```

```
plgrp <- list(
  "MA" = c("MA","LH","SD"),
  "KU" = c('KU','NU','TO','TS'),
  "KS" = c('KS','PN','TL','EL'),
  "HB" = c('HB','PP'),
  "LM" = c("LM"),
  "LV" = c("LV"),
  "XX" = c("TA","SA","VA","JA","KP")
)
```

```
puuliike=length(plgrp)
```

```
# Siia salvestame mudelite abil prognoositud puude kõrgused (NB! mudeli hindamisel pole selle
puu kõrgust kasutatud):
```

```
# prognoosid kasutades kõigi varasemate aastate andmeid:
```

```
prognoos_koik=rep(NA, dim(valim1)[1])
```

```
# prognoosid kasutades kõigist viimasel 5 aastal kogutud andmeid:
```

```
prognoos_5a =rep(NA, dim(valim1)[1])
```

```
# jagame kõik traktid 20-sse gruppi. Hiljem jätame ühes grupis (5% traktidest) mõõdetud puud
mudeli hindamisest välja
```

```
# hindame mudeli(d) vaid 95% andmete pealt ja vaatame, kui hästi nad väljajäänud osa
prognoosivad:
```

```
n_traktigruppe=20
```

```
ntrakte = length(table(valim1$atrakt_id))
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
set.seed(1)
grupid = sample(1:n_traktigruppe, ntrakte, replace=TRUE)
traktinimed=names(table(valim1$atrakt_id))

ajut=rep(NA, dim(valim1)[1])

for (i in 1:ntrakte){
  ajut[valim1$atrakt_id == traktinimed[i] ] = grupid[i]
}

valim1$traktigrupp=ajut

# Kontrolliks:
# valim1[1:10,]

# Eemaldame ühe grupi valimist, seejärel leiame mudeli a) kasutades kõiki andmeid kuni (kaasa
arvatud) prognoositav aasta vs vaid 5 aasta andmeid

for ( eemaldagrupp in 1:n_traktigruppe){

valim2=valim1[valim1$traktigrupp!=eemaldagrupp,]

for ( prognoositav_aasta in (1999+4):max(valim2$aasta) ){

mudelid=list(puuliike)

# Valime puuliigi
for (xpl in 1:puuliike){
# xp1=3
print("Koostan mudeleid järgmiste puuliikide jaoks:")
print(plgrp[[xpl]])

# Mudelid juhul kui kasutan kogu hetkeks olemasolevat ajalaugu: 1999... prognoositava aastani
(viimane kaasa arvatud)
mudeli_ajavahemik=1999:prognoositav_aasta

pars=NULL
nls.tul=NULL

valim <- valim2[valim2$puuliik %in% plgrp[[xpl]] & valim2$aasta %in% mudeli_ajavahemik,]
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# Hindame mudeli kasutades moodustatud andmestikku valim:
```

```
# -----
```

```
kkpars<- function(b1, c1, c2, h, d, h100, onsaar)
{
  x <- d/(d+b1)
  y <- x^(c1+h100*c2)
  w <- (h-1.3)/y
  lm.t <- lm(w~h100+onsaar)
  cf <-<- coef(lm.t)
  1.3+(cf[[1]+cf[[2]*h100+cf[[3]*onsaar] *(d/(d+b1))^(c1+h100*c2)
}
```

```
# Kõrguskõvera parameetrite arvutamine kahe-astmelisena
```

```
cf <- NULL
```

```
nls.tul <- gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim,
start=list(b1=4, c1=2, c2=0.1), algorithm = "lmaccel")
```

```
# gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim,
start=list(b1=4, c1=2, c2=0.1), trace=TRUE)
```

```
# if(inherits(nls.tul, "try-error")) {tulvek[i]=NA; print("Probleemid...")} else {
```

```
  pars <- as.list(nls.tul$m$getPars())
```

```
  pars$a1 = cf[[1]]
```

```
  pars$a2 = cf[[2]]
```

```
  pars$a3 = cf[[3]]
```

```
  nls.tul.2 <- gsl_nls( h_puu/10
~1.3+(a1+a2*arv_h100+a3*onsaar)*((d_13/10)/((d_13/10)+b1))^(c1+arv_h100*c2), valim,
start=pars, algorithm="lmaccel")
```

```
# }
```

```
mudelid[[xpl]] = nls.tul.2
```

```
} # tsükkel üle puuliikide
```

```
# Leiame prognoosid mudeli hindamiseks mittekasutatud puudele:
```

```
for (i in 1:puuliike){
```

```
  indeks=valim1$puuliik %in% plgrp[[i]] & valim1$aasta==prognoositav_aasta &
valim1$straktigrupp==eemaldagrupp
```

```
  valim <- valim1[indeks,]
```

```
  prognoos_koik[indeks]=predict(mudelid[[i]], valim)
```

```
}
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
} # prognoositav_aasta
} # traktigrupp

# Sama arvutus prognoosidele, kui kasutame vaid viimase viie aasta andmeid prognoosi
leidmiseks:

for ( eemaldagrupp in 1:n_traktigruppe){

valim2=valim1[valim1$traktigrupp!=eemaldagrupp,]

for ( prognoositav_aasta in (1999+4):max(valim2$aasta) ){

mudelid=list(puuliike)

# Valime puuliigi
for (xpl in 1:puuliike){
# xp1=3
print("Koostan mudeleid järgmiste puuliikide jaoks:")
print(plgrp[[xpl]])

# Mudelid juhul kui kasutan vaid viimase viie aasta andmeid (viimane kaasa arvatud)
# -----
# NB! Muutus võrreldes eelmise programmilõiguga:
# -----
mudeli_ajavahemik=(prognoositav_aasta-4):prognoositav_aasta
# -----

pars=NULL
nls.tul=NULL

valim <- valim2[valim2$puuliik %in% plgrp[[xpl]] & valim2$aasta %in% mudeli_ajavahemik,]

# Hindame mudeli kasutades moodustatud andmestikku valim:
# -----

kkpars<- function(b1, c1, c2, h, d, h100, onsaar)
{
  x <- d/(d+b1)
  y <- x^(c1+h100*c2)
  w <- (h-1.3)/y
  lm.t <- lm(w~h100+onsaar)
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
cf <- coef(lm.t)
1.3+(cf[1]+cf[2]*h100+cf[3]*onsaar)*(d/(d+b1))^(c1+h100*c2)
}

# Kõrguskõvera parameetrite arvutamine kahe-astmelisena
cf <- NULL
nls.tul <- try(gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim,
start=list(b1=4, c1=2, c2=0.1), algorithm = "lmaccel"), silent=TRUE)

# gsl_nls( h_puu/10 ~kkpars(b1, c1, c2, h_puu/10, d_13/10, arv_h100, onsaar), valim,
start=list(b1=4, c1=2, c2=0.1), trace=TRUE)

# if(inherits(nls.tul, "try-error")) {tulvek[i]=NA; print("Probleemid...")} else {
pars <- as.list(nls.tul$m$getPars())
pars$a1 = cf[[1]]
pars$a2 = cf[[2]]
pars$a3 = cf[[3]]
nls.tul.2 <- gsl_nls( h_puu/10
~1.3+(a1+a2*arv_h100+a3*onsaar)*((d_13/10)/((d_13/10)+b1))^(c1+arv_h100*c2), valim,
start=pars, algorithm="lmaccel")
# }

mudelid[[xpl]] = nls.tul.2
} # tsükkel üle puuliikide

# Leiame prognoosid mudeli hindamiseks mittekasutatud puudele:

for (i in 1:puuliike){
indeks=valim1$puuliik %in% plgrp[[i]] & valim1$aasta==prognoositav_aasta &
valim1$traktigrupp==eemaldagrupp
valim <- valim1[indeks,]

# -----
# NB! Muutus võrreldes eelmise programmilõiguga:
# -----
prognoos_5a[indeks]=predict(mudelid[[i]], valim)
# -----

}

} # prognoositav_aasta
} # traktigrupp
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
# Leiame prognoosivead:
viga_5a= valim1$h_puu/10-prognoos_5a
viga_koik= valim1$h_puu/10-prognoos_koik

# Leiame RMSE
sqrt(mean(viga_5a**2, na.rm=T))
sqrt(mean(viga_koik**2, na.rm=T))
# > sqrt(mean(viga_5a**2, na.rm=T))
# [1] 2.56934
# > sqrt(mean(viga_koik**2, na.rm=T))
# [1] 2.597064

# Prognoosivigade keskmised aasta kaupa
nihked_5a=as.vector(by(viga_5a, valim1$aasta, mean))
nihked_koik=as.vector(by(viga_koik, valim1$aasta, mean))

# > nihked_5a
# [1] NA NA NA NA 0.168513811 0.005623734 0.053118889 -0.138169575
0.515744577 0.747310932 0.337812665 0.316847626
# [13] 0.046408740 0.023499753 0.208440164 -0.095975937 0.079242148 -0.164247763 -
0.599304717 -0.347189658 -0.237212251 -0.171003686 -0.107879493
# > nihked_koik
# [1] NA NA NA NA 0.16851381 0.04572700 0.11399404 -0.06048379
0.62733065 0.92747565 0.63755705 0.73545664 0.62925637
# [14] 0.62475067 0.76677886 0.42938688 0.56670292 0.27826141 -0.26587993 -
0.17569851 -0.14359874 -0.25626736 -0.31599029

# Joonis 13 - nihked kõrguse hindamisel aasta kaupa
windows(width=4*3, height=3*3)
plot(1999:2021, nihked_koik, ylim=c(-1,1), xlim=c(1999+4, 2021), pch=21, bg="pink", col="red",
xlab="Aasta", cex=1.5, ylab="Keskmine viga puu kõrguse hindamisel, m", cex.axis=1.2,
cex.lab=1.2)
lines(1999:2021, nihked_koik, col="#FFD0D0")

lines(1999:2021, nihked_5a, col="gray80")
points(1999:2021, nihked_5a, bg="skyblue", col="navyblue", pch=21, cex=1.2)

abline(h=0, lwd=3)

legend(2003, -0.5, c("kõiki andmeid", "viimase 5a andmeid"), title="Kõrguskõvera mudeli \n
hindamisel kasutati:", pch=21, col=c("red","navyblue"), pt.bg=c("pink","skyblue"), pt.cex=1.5,
bg="white", cex=1.1, bty="n", title.adj=0)
```


Versioon 2.0
Kalev Pärna 31.05.2024



Joonis 14 - RMSE väärtused aastati

```
RMSE_5a=sqrt(as.vector(by(viga_5a**2, valim1$aasta, mean)))
RMSE_koik=sqrt(as.vector(by(viga_koik**2, valim1$aasta, mean)))

plot(1999:2021, RMSE_koik, xlim=c(1999+4, 2021), pch=21, bg="pink", col="red", xlab="Aasta",
     cex=1.5, ylab="Viga üksikuude kõrguse prognoosimisel, RMSE", cex.axis=1.2, cex.lab=1.2)
lines(1999:2021, RMSE_koik, col="#FFD0D0")

lines(1999:2021, RMSE_5a, col="gray80")
points(1999:2021, RMSE_5a, bg="skyblue", col="navyblue", pch=21, cex=1.2)

legend(2015, 2.5, c("kõiki andmeid", "viimase 5a andmeid"), title="Kõrguskõvera mudeli \n
hindamisel kasutati:", pch=21, col=c("red","navyblue"), pt.bg=c("pink","skyblue"), pt.cex=1.5,
bg="white", cex=1.1, bty="n", title.adj=0)
```

4.5. Optimaalse ajaakna leidmine mahumudeli jaoks

4.5.1. Analüüsi skeem

Eesmärk on leida optimaalne ajaaken puistu tagavara hindamiseks – kui pikka eelnevat ajaintervalli peaks mahumudeli kalibreerimiseks kasutama, et mudeli abil saadud prognoosid oleksid võimalikult täpsed. Siinkohal me peame silms mahumudelit, mis on toodud SMI Arvutusmetoodilises juhendis (lk. 11, valem (3)):

$$M = (a_1 + a_2 \cdot H100 + a_3 \cdot OnSaar) \left(\frac{A}{A+b_1} \right)^{c_1+c_2 \cdot H100},$$

kus H100 on kasvukoha headuse indeks (boniteet),
A on vanus ('keskmise' puu järgi antud visuaalne hinnang),
OnSaar on 0-1 tunnus.

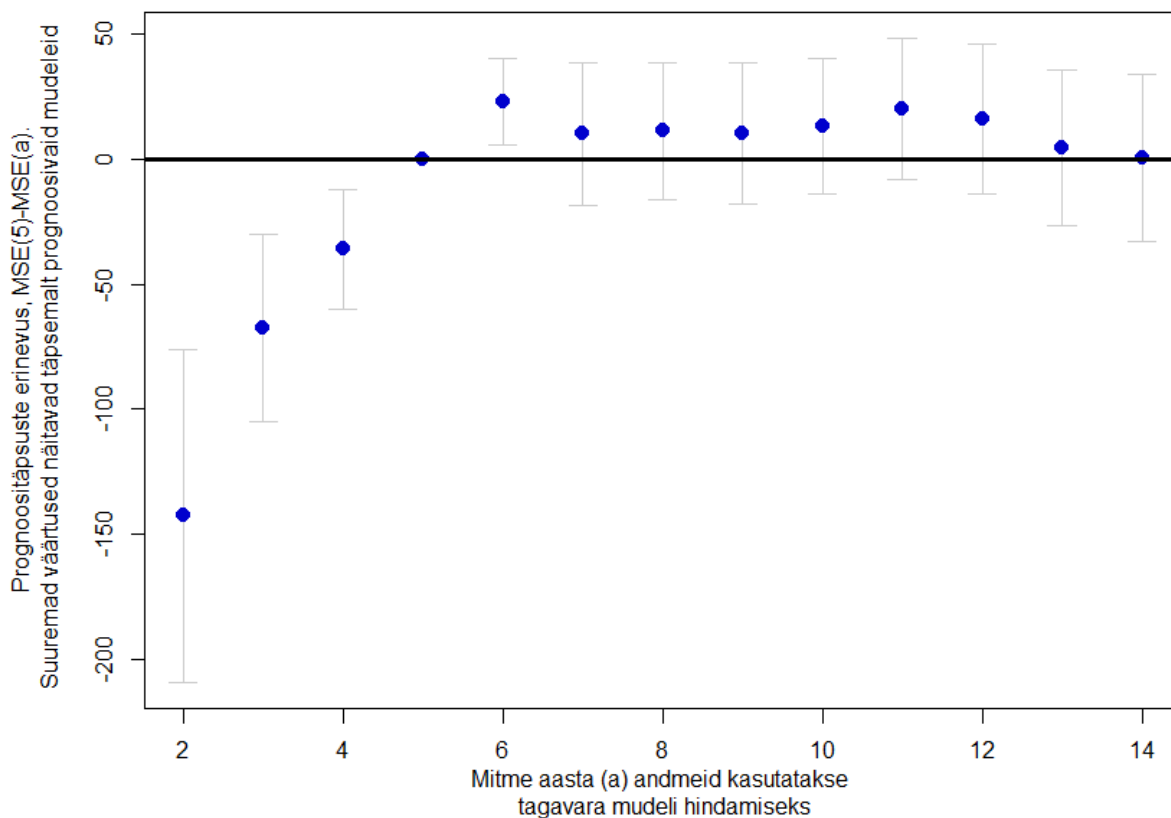
Mugavuse mõttes tähistame selle mudeli **M(H100, A)**.

Parameetri b1 loeme fikseerituks: b1=10. Sisulist muutust ei tohiks see kaasa tuua, sest tegemist on peaaegu üleparametriseeritud mudeliga - väga laiades piirides antud parameetri b1 väärtust muutes kohanevad teised parameetrid nii, et prognoosid jäävad samaks.

Mahumudeli M(H100, A) kalibreerimisel on alljärgnevas analüüsis kasutatud alalistel ja ajutistel traktidel asuvaid tagavaraproovitükke, mis esinevad andmetes jaotamata kujul. Nimetatud proovitükkidel on SMI andmestikus mahutunnus *arv_maht_es* (klupitud puude mahtude summa), mida me antud uurimuses käsitleme kui '**tõde**'. Kõrguse *h* mudel on kogu analüüsi vältel fikseeritud sellisel kujul nagu teda on kasutatud mahutunnuse *arv_maht_es* ('tõde') arvutamisel. Samuti on fikseeritud H100 mudel.

Mahumudeleid hindame vaid neid maatükke kasutades, kus maakategooria on kas M või MM, prtliik=1 (tagavara) ja kus proovitükk osadeks jagamata (pindosak=1). Lisaks peab kehtima *on_sees*="J" ja proovitükil peab olema fikseeritud metsa vanus, vanus > 0.

4.5.2. Tulemused ja kokkuvõte



Joonis 15

Antud pilt on leitud ristvalideerimise meetodil: mõõdetakse nende proovitükide prognoosi täpsust, mida pole mudeli hindamisel kasutatud. Täpsemalt väljendudes, iga kord eemaldatakse 5% traktidest, ülejäänud 95% traktide pealt hinnatakse mahumudel, mille abil prognoositakse mudeli hindamisest väljajäänud 5% traktide mahtusid. Nii toimitakse 20 korda, millega kaetakse kõik proovitükid.

Antud pildi joonistamiseks on kasutatud vaid neid proovitükke, mille jaoks on võimalik kasutada kuni 14 aasta pikkust ajalugu. Seega kõige vanemad analüüsi alla minevad proovitükid pärinevad aastast 2012.

Kokkuvõte: Jooniselt 15 nähtub, et tagavara prognoosimisel mudeliga M(H100, A) (SMI arvutusmetoodiline juhend, lk 11 valem (3)) on optimaalne kasutada 6-aastast ajaakent st kui mahumudeli kalibreerimist teostada eelneva 6 aasta andmete põhjal, siis saadavad prognoosid vastavad kõige enam tegelikele mahtudele. See reegel peab paika enamikel aastatel, kuid pikema perioodi kestel võib esineda ka üksikuid erandeid.

Version 2.0
Kalev Pärna 31.05.2024



4.5.3. Programmi kood

Tagavara prognoos - kui pikka ajalugu vajame mudeli hindamisel

Algandmete sisselugemine

```
library("RSQLite")
```

```
library(gslnls)
```

```
## connect to db
```

```
con <- dbConnect(drv=RSQLite::SQLite(), dbname="smi_koik.sqlite")
```

```
## list all tables
```

```
tables <- dbListTables(con)
```

```
## exclude sqlite_sequence (contains table information)
```

```
tables <- tables[tables != "sqlite_sequence"]
```

```
## create a data.frame for each table
```

```
for (i in seq(along=tables)) {
```

```
  assign(tables[i], dbGetQuery(conn=con, statement=paste("SELECT * FROM ", tables[[i]], "",  
sep="")))
```

```
}
```

```
# Ametlik variant (ei sisalda "põõsaid"),
```

```
# gruppi XX lisatud ka "puuliik" XX -- siis töötab programm samamoodi nii siis, kui sisendiks  
puuliik kui siis, kui juba eelnevalt grupiks teisedatud.
```

```
plgrp <- list(  
  "MA" = c("MA", "LH", "SD"),  
  "KU" = c('KU', 'NU', 'TO', 'TS'),  
  "KS" = c('KS', 'PN', 'TL', 'EL'),  
  "HB" = c('HB', 'PP'),  
  "LM" = c("LM"),  
  "LV" = c("LV"),  
  "XX" = c("TA", "SA", "VA", "JA", "KP", "XX")  
)
```

```
puuliike=length(plgrp)
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
valim1 <- prtosa[prtosa$on_sees=="J" & prtosa$prtliik==1 & prtosa$maakategooria %in%  
c("M", "MM") & prtosa$pindosak==1 & prtosa$vanus>0,]
```

```
maxaastaid=length(table(valim1$aasta))  
maxaastaid
```

```
# Siia salvestame mudelite abil prognoositud tagavarad (NB! mudeli hindamisel pole  
prognoositava proovitüki osa enda andmeid kasutatud):  
# Esimene tulp - antud tüki prognoos sama aasta andmete pealt; teine tulp - prognoos käesoleva  
aasta ja eelmise aasta põhjal jne.
```

```
prognoos_maatriks=matrix(NA, nrow=dim(valim1)[1], ncol=maxaastaid)  
koondumine = matrix(NA, nrow=dim(valim1)[1], ncol=maxaastaid)
```

```
# jagame kõik traktid 20-sse gruppi. Hiljem jätame ühes grupis (5% traktidest) mõõdetud puud  
mudeli hindamisest välja
```

```
# hindame mudeli(d) vaid 95% andmete pealt ja vaatame, kui hästi nad väljajäänud osa  
prognoosivad:
```

```
n_traktigruppe=20
```

```
ntrakte = length(table(valim1$atrakt_id))  
set.seed(1)  
grupid = sample(1:n_traktigruppe, ntrakte, replace=TRUE)  
traktinimed=names(table(valim1$atrakt_id))
```

```
puu_traktigrupi_number=rep(NA, dim(valim1)[1])
```

```
for (i in 1:ntrakte){  
  puu_traktigrupi_number[valim1$atrakt_id == traktinimed[i] ] = grupid[i]  
}
```

```
valim1$traktigrupp=puu_traktigrupi_number
```

```
# Kontrolliks:  
# valim1[1:10,]
```

```
# Eemaldame ühe grupi valimist, seejärel leiame mudeli a) kasutades kõiki andmeid kuni (kaasa  
arvatud) prognoositav aasta vs vaid 5 aasta andmeid
```

```
for ( eemaldagrupp in 1:n_traktigruppe){  
  print (paste("Traktigrupp:", eemaldagrupp, "/", n_traktigruppe ))
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
valim2=valim1[valim1$traktigrupp!=eemaldagrupp,]

for ( prognoositav_aasta in (1999):(1999+maxaastaid-1) ){
print (paste("Prognoosin aastat:", prognoositav_aasta ))
flush.console()

for (aastaid_mudelis in 1:(prognoositav_aasta-1999+1)){

# Siia salvestame prognoosivad mudelid - iga puuliigi(grupi) jaoks üks mudel
mudelid=list(puuliike)
# Siia läheb kirja informatsioon selle kohta, kas hinnatud mudel koondus
koondumine_abi=rep(NA, puuliike)

# Valime puuliigi
for (xpl in 1:puuliike){

# Mudeli loomiseks kasutatav ajavahemik:
mudeli_ajavahemik=(prognoositav_aasta-aastaid_mudelis+1):prognoositav_aasta

nls.tul=NULL

# Nopime mudeli hindamiseks välja andmestiku - vaid sobiv enamuspuuliik (arv_epl) ja
takseerimisaja kuulub soovitud vahemikku
valim <- valim2[valim2$arv_epl %in% plgrp[[xpl]] & valim2$aasta %in% mudeli_ajavahemik,]

# Hindame mudeli kasutades moodustatud andmestikku valim:
# -----
# -----
# Variant 1
# -----

mahufunktsioon <- function(a1, a2, b1, c1, c2, h100, vanus)
{
M=(a1+a2*h100)*(vanus/(vanus+b1))^(c1+c2*h100)
return(M)
}

nls.tul <- gsl_nls( arv_maht_es ~mahufunktsioon(a1,a2,b1=10,c1,c2, arv_h100, vanus), valim,
start=list(a1=-100, a2=30, c1=16, c2=-0.2), algorithm = "lmaccel", control =
gsl_nls_control(maxiter = 2000))

# Kui mudel ei koandu, siis proovi teiste algühenditega
if (!nls.tul$converInfo$isConv)
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
nls.tul <- gsl_nls( arv_maht_es ~mahufunktsioon(a1,a2,b1=10,c1,c2, arv_h100, vanus), valim,
  start=list(a1=-380, a2=35, c1=16, c2=-0.2), algorithm = "lmaccel", control =
gsl_nls_control(maxiter = 5000))
if (!nls.tul$convInfo$sisConv)
  nls.tul <- gsl_nls( arv_maht_es ~mahufunktsioon(a1,a2,b1=10,c1,c2, arv_h100, vanus), valim,
  start=list(a1=-34, a2=23, c1=20, c2=-0.3), algorithm = "lmaccel", control =
gsl_nls_control(maxiter = 5000))
if (!nls.tul$convInfo$sisConv)
  nls.tul <- gsl_nls( arv_maht_es ~mahufunktsioon(a1,a2,b1=10,c1,c2, arv_h100, vanus), valim,
  start=list(a1=-70, a2=24, c1=10, c2=-0.2), algorithm = "lmaccel", control =
gsl_nls_control(maxiter = 5000))
```

```
# -----
# Variant 1
# -----
```

```
# Kui ikka ei koondunud, siis anna veahäire!
```

```
koondumine_abi[xpl]=nls.tul$convInfo$sisConv
if (!nls.tul$convInfo$sisConv) {
  print("Koondumisprobleem!");
  print(paste(xpl, ":", paste(plgrp[[xpl]], collapse=",")));
  print(eemaldagrupp)
  print(proгноositav_aasta)
  print(mudeli_ajavahemik)
  print(aastaid_mudelis)
  stop("Koondumisprobleem!!")
}
```

```
mudelid[[xpl]] = nls.tul
```

```
} # lõppeb tsükkel üle puuliikide
```

```
# Leiame prognoosid mudeli hindamiseks mittekasutatud puudele:
```

```
for (i in 1:puuliike){
  indeks=valim1$arv_epl %in% plgrp[[i]] & valim1$aasta==prognoositav_aasta &
valim1$traktigrupp==eemaldagrupp
  valim <- valim1[indeks,]
  prognoos_maatriks[indeks,aastaid_mudelis]=predict(mudelid[[i]], valim)
  koondumine[indeks, aastaid_mudelis] = koondumine_abi[[i]]
}
```

Versioon 2.0
Kalev Pärna 31.05.2024



```
} # lõppeb tsükkel: aastaid mudelis
} # lõppeb tsükkel: prognoositav_aasta
} # lõppeb tsükkel: traktigrupp

#-----

# Leiame prognoosivead (tegelik arvutatud klupitud puude järgi - mudeli prognoos):

veamaatriks=matrix(NA, nrow=dim(prognoos_maatriks)[1], ncol=dim(prognoos_maatriks)[2])
for (i in 1:maxaastaid){
  veamaatriks[,i]=valim1$arv_maht_es-prognoos_maatriks[,i]
}

# Paarikaupa sõltuvate valimite t-testid: kas tagavara prognoosis paremini 1 aasta mudel vs 5
aasta mudel jne:

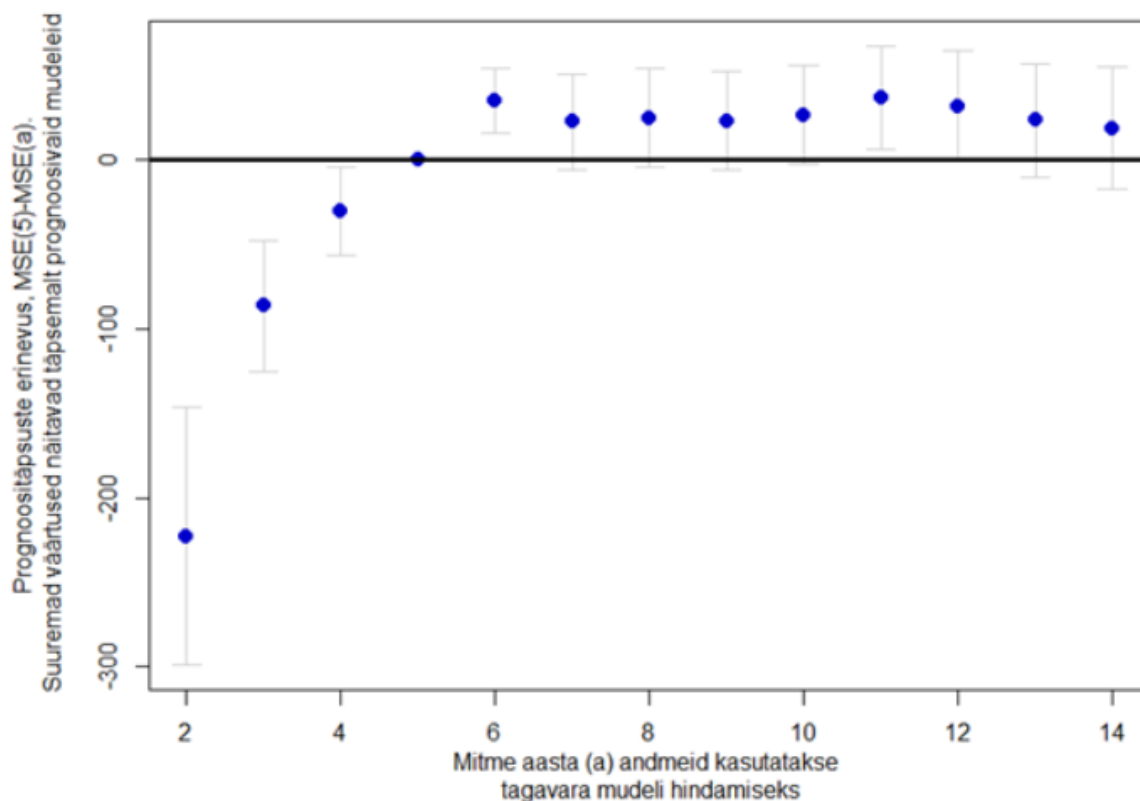
kaugeajalugu=14
ind=(valim1$aasta-1999+1)>=kaugeajalugu

hinnang=rep(NA, kaugeajalugu)
UIalumine=rep(NA, kaugeajalugu)
UIylemine=rep(NA, kaugeajalugu)

for (i in 1:kaugeajalugu){
  testitulemus=t.test(veamaatriks[ind,5]**2-veamaatriks[ind,i]**2)
  hinnang[i]=testitulemus$estimate
  UIalumine[i]=testitulemus$conf.int[1]
  UIylemine[i]=testitulemus$conf.int[2]
}

# Tulemuste visualiseerimine
windows(width=4*3, height=3*3)
par(mar=c(5.5,5.5,3.1,2.1))
plot(2:kaugeajalugu, hinnang[-1], ylim=range(c(UIalumine[-1], UIylemine[-1])), na.rm=T),
xlab="Mitme aasta (a) andmeid kasutatakse \n tagavara mudeli hindamiseks",
ylab="Prognoositäpsuste erinevus, MSE(5)-MSE(a).\n Suuremad väärtused näitavad täpsemalt
prognoosivaid mudeleid")
arrows(2:kaugeajalugu, UIalumine[-1], 2:kaugeajalugu, UIylemine[-1], angle=90, length=0.1,
code=3, col="gray80")
points(2:kaugeajalugu, hinnang[-1], pch=20, col="blue3", cex=2)
abline(h=0, lwd=3)
```


Lisame lõppu analoogse joonise mudeli variandist, kus ka tunnus *OnSaar* on sisendina kasutusel (täpselt nagu SMI arvutusjuhendis - lk 11, valem (3)), parameeter b_1 on endiselt fikseeritud, $b_1=10$. Parimate ajaakendena tulevad nüüd esile 6 ja 11 aastat.



Joonis 16

Kommentaari: Mahumudeli $M(H_{100}, A)$ jaoks optimaalse ajaakna leidmine on võrreldes kõrgusmudeliga mõnevõrra komplitseeritud ülesanne. Põhjuseks on nähtavasti mahumudeli üldiselt suur prognoosiviga, sest tema sisendite hulka ei kuulu ühtki puistu tiheduse näitajat. Suhteliselt lihtsalt tuleb analüüsist välja küll see statistika põhitõde, et liiga lühikese (lühema kui 5 aastat) kalibreerimisperioodi kasutamine teeb juhusliku prognoosivea veelgi suuremaks (sest vaatluste arv väheneb), kuid pikemate ajaakende võrdluses 6 aastase andmeintervalli paremus teiste ees ei paista olevat eriti suur.

5. Kasutatud kirjanduse loetelu

1. Statistiline metsainventeerimine: Arvutusmetoodiline juhend. Keskkonnaministeerium, 2018.
2. Statistilise metsainventuuri (SMI) traktivõrgu analüüs ja arvutusmeetoditega seotud uuring. TÜ MSI projektaruanne. Keskkonnaministeerium, 2023.
3. Breiman, L. (2001). Random Forests. *Machine Learning*, **45** (1), 5–32.
4. Deming, W. E., and Stephan, F. F. (1940), On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Tables Are Known. *The Annals of Mathematical Statistics*, **11**, 427- 44.
5. IPCC (2006). 2006 IPCC Guidelines for National Greenhouse Gas Inventories. Vol. 4: Agriculture, Forestry and Other Land Use. Prepared by the National Greenhouse Gas Inventories Programme, Eggleston H.S., Buendia L., Miwa K., Ngara T. and Tanabe K. (eds). Published: IGES, Japan.
6. Little, R. J., Wu, M. M. (1991) Models for contingency tables with known margins when target and sampled populations differ. *Journal of the American Statistical Association*, **86** (413), 87-95.
7. Tomppo E., Gschwantner Th., Lawrence M., McRoberts R.E. (eds.). (2010). In: *National forest inventories: pathways for common reporting*. Springer. 612 p. <http://dx.doi.org/10.1007/978-90-481-3233-1>.
8. Zhang, P. (2003), Multiple Imputation: Theory and Method. *International Statistical Review / Revue Internationale de Statistique*, **71** (3), 581-592.

6. Lisa

Aruandele on lisatud elektroonsel kujul järgmised failid:

- 1) kasutusvalmis LULUCF muutuste tabelid .xlsx formaadis (kokku 14 .xlsx faili ehk 2 faili iga maakategooria kohta, milles üks põhineb ainult alaliste proovitükkide infol, teine aga kasutab nii alaliste kui ka ajutiste proovitükkide infot).

lcfmk_C_ajutistega.xlsx

lcfmk_C_uus3.xlsx

lcfmk_F_ajutistega.xlsx

lcfmk_F_uus3.xlsx

lcfmk_G_ajutistega.xlsx

lcfmk_G_uus3.xlsx

lcfmk_O_ajutistega.xlsx

lcfmk_O_uus3.xlsx

lcfmk_P_ajutistega.xlsx

lcfmk_P_uus3.xlsx

lcfmk_S_ajutistega.xlsx

lcfmk_S_uus3.xlsx

lcfmk_templ2.xlsx

lcfmk_W_ajutistega.xlsx

lcfmk_W_uus3.xlsx

- 2) Aruande LULUCF osa lugemist hõlbustav fail

Lulucf_4.html

(kvaliteetsem tabelite vormistus).

- 3) Aruande LULUCF osa R-koodid:

Lulucf_4.Rmd (koodi põhiosa)

proovitykiosade_koordinaatide_arvestamine.R (täiendav kood, mille poole programmis pöördatakse käsuga *include*).

Versioon 2.0
Kalev Pärna 31.05.2024



Aruande koostaja: **Kalev Pärna**

Aruande valmimisse panustasid:

Raul Kangro, Märt Möls (analüüsid ja arvutuseksperimendid)

Imbi Traat, Meelis Käärrik (konsultatsioonid)

31.05.2024

Kalev Pärna

TÜ MSI emeriitprofessor

Projekti vastutav täitja