

## Liite 2 / Bilaga 2

### Kokeilukuntien otannan ohjelmointikomennot

#### Programmeringskommandona för urvalet av försökskommuner

**Kunnat, joiden sisältä valitaan myöhemmin kaksivuotista esiopetusta järjestävät kokeilutoimipaikat ja niiden verrokkitoimipaikat:**

**Kommuner inom vilka man senare väljer de försöksverksamhetsställen som ska ordna tvåårig förskoleundervisning och kontrollverksamhetsställena:**

```
# Code author:
# Ramin Izadi
# ramin.izadi@aalto.fi

rm(list=ls())
library(tidyverse)
library(haven)

sessionInfo()
set.seed(NULL)

kunnat = read_csv("final_sample/kunnat.csv")

# Imputoinnit (maksu menetys, vaka_n16, eivaka_n16) -----
-----

# Predict number of children based number of daycare centers, population
and education
impute_model_vaka = glm(vaka_n16 ~ yksikot_vaka_oph + yksikot_oph +
vakiluku18 + I(vakiluku18^2) +
                        osuus_korkea_aste + osuus_toinen_aste +
osuus_ulkomaalaistaustaiset +
                        as.factor(alueluokka), family = poisson, data =
kunnat %>% filter(vakiluku18 < 50000))
impute_model_eivaka = glm(eivaka_n16 ~ yksikot_vaka_oph + yksikot_oph +
vakiluku18 + I(vakiluku18^2) +
                        osuus_korkea_aste + osuus_toinen_aste +
osuus_ulkomaalaistaustaiset +
                        as.factor(alueluokka), family = poisson, data
= kunnat %>% filter(vakiluku18 < 50000))

impute_model_vaka %>% summary

# Add imputations when original is NA
kunnat = kunnat %>% mutate(vaka_n16_impute = predict(impute_model_vaka,
kunnat, type = "response"),
```

```

vaka_n16 = ifelse(is.na(vaka_n16),
vaka_n16_impute, vaka_n16),
eivaka_n16_impute =
predict(impute_model_eivaka, kunnat, type = "response"),
eivaka_n16 = ifelse(is.na(eivaka_n16),
eivaka_n16_impute, eivaka_n16))

# Imputoi maksumenetys muiden kuntien keskiarvolla
kunnat = kunnat %>% mutate(maksumenetys = ifelse(is.na(maksumenetys) |
maksumenetys == 0,
mean(maksumenetys, na.rm
= T), maksumenetys))

kunnat %>% select(nimi, vakiluku18, vaka_n16, vaka_n16_impute,
eivaka_n16, eivaka_n16_impute)

# Lisää muuttujia -----
----

kunnat = kunnat %>%
  filter(nimi != "Tampere") %>%
  mutate(yksikot = ifelse(is.na(yksikot), yksikot_oph, yksikot),
kelpoiset_yksikot_osuus = yksikot/yksikot_vaka_oph,
kelpoiset_yksikot_osuus = ifelse(kelpoiset_yksikot_osuus > 1, 1,
kelpoiset_yksikot_osuus)) %>%
  mutate(lapset16 = eivaka_n16 + vaka_n16*kelpoiset_yksikot_osuus,
koti_osuus16 = eivaka_n16/lapset16) %>%
  mutate(vaka_kustannus16 = (50 + 30 + kuljetus_kustannus18_OPH +
maksumenetys*0.6*9.5) * vaka_n16 * kelpoiset_yksikot_osuus,
eivaka_kustannus16 = (50 + 30 + esiopetus_kustannus18_OPH) *
eivaka_n16,
verrokki_kustannus16 = 50 * lapset16,
odotettu_kustannus16 = vaka_kustannus16 + eivaka_kustannus16 +
verrokki_kustannus16) %>%
  mutate(yksikot_kori = cut(yksikot,
breaks = c(0,20,50,200,300),
label = c(20,50,200,300)),
yksikot_kori = as.numeric(as.character(yksikot_kori)))

# Kuntakorit -----
--

#c("Kempele", "Kauhava-Evijärvi-Lappajärvi", "Masku", "Varkaus",
"Hollola")
ylisuuret_kunnat = kunnat %>% filter(yksikot >= 6, yksikot_oph <= 4) %>%
pull(nimi)
ylisuuret_kunnat
ylipienet_kunnat = kunnat %>% filter(yksikot <= 4, yksikot_oph >= 5) %>%
pull(nimi)
ylipienet_kunnat

kunnat = kunnat %>%
  mutate(yksikot_kori = ifelse(alueluokka == "kaupunki" & avi != "Lapin
AVI", yksikot_kori, "harva")) %>%

```

```

unite("kori_temp", avi, yksikot_kori, sep = ".", remove = F) %>%

# Käsitehdyt muutokset koreihin
mutate(kelvoton = yksikot_oph == 0 | yksikot == 0,
       kori = ifelse(kieli == "ruotsi", "Ruotsinkielinen", kori_temp),
# siirrä tälle yhden alemmas jo haluat että pienet ruotsinkieliset
kunnat menevät myös "ruotsi" koriin
       kori = ifelse(yksikot_oph %in% 1:4 | nimi %in% ylipienet_kunnat,
"pieni", kori),
       kori = ifelse(kelvoton == T, NA, kori),
       kori = ifelse(nimi %in% ylisuuret_kunnat, kori_temp, kori)) %>%

# Lisää suurimman kunnan nimi selvyuden vuoksi
group_by(kori) %>%
mutate(suurin_kunta = nimi[vakilukul8 == max(vakilukul8)]) %>% ungroup

save(kunnat, file = "final_sample/kunnat_augmented.Rdata")

# Pienten kuntien kustannukset -----
--

# Keskimääräinen pienten kuntien kustannus (odotusarvo, kun 50%
koeryhmässä)
kunnat_pienet_kustannukset = kunnat %>% filter(kori == "pieni") %>%
  summarise(kokeilu_kustannus = sum(vaka_kustannus16 + eivaka_kustannus16
+ verrokki_kustannus16)/2,
           vaka_n = sum(vaka_n16*kelpoiset_yksikot_osuus)/2,
           eivaka_n = sum(eivaka_n16)/2,
           verrokki_n = sum(lapset16)/2,
           koti_osuus16 = sum(eivaka_n16)/sum(lapset16))

kunnat_pienet_kustannukset

# Suurten kuntien otanta -----
-----

allokaatio_suhde = 0.20 # allocation share
kuntien_osuus = 0.65

puskuri = 1000000
budjetti = 14000000 - puskuri -
kunnat_pienet_kustannukset$kokeilu_kustannus
budjetti

kunnat_suuret = kunnat %>%
  filter(kori != "pieni") %>%
  mutate(kokeilu_kustannus =
           allokaatio_suhde * vaka_kustannus16 +
           allokaatio_suhde * eivaka_kustannus16 +
           (1-allokaatio_suhde) * verrokki_kustannus16)

# Rerandimisoi kunnes budjetti alitetaan
kustannus = Inf
while(kustannus > budjetti){
  kunnat_suuret_otos = kunnat_suuret %>%
    group_by(kori) %>%

```

```

    sample_n(round(kuntien_osuus*n())) %>% ungroup

    kustannus = sum(kunnat_suuret_otos$kokeilu_kustannus)
    print(kustannus)
}

# Kokonaiskustannukset suurille kunnille
kunnat_suuret_kustannukset = kunnat_suuret_otos %>%
  summarise(kokeilu_kustannus = sum(kokeilu_kustannus),
            vaka_n =
sum(vaka_n16*kelpoiset_yksikot_osuus)*allokaatio_suhde,
            eivaka_n = sum(eivaka_n16)*allokaatio_suhde,
            verrokki_n = sum(lapset16)*(1-allokaatio_suhde),
            koti_osuus16 = sum(eivaka_n16)/sum(lapset16))

# Arvioitu kokonaiskustannus ja osallistuvien lasten määrät
kustannukset = bind_rows(kunnat_pienet_kustannukset,
kunnat_suuret_kustannukset)
kustannukset = kustannukset %>% add_row(kunnat_pienet_kustannukset +
kunnat_suuret_kustannukset)
kustannukset %>% select(-koti_osuus16)

# Lopullinen otos:
kunnat_suuret_otos_lopullinen = kunnat_suuret_otos %>% select(nimi,
kunta, kori)
write_excel_csv2(kunnat_suuret_otos_lopullinen,
"final_sample/suurten_kuntien_lopullinen_otos.csv")

```

**Kunnat, jotka on jaettu joko koe- tai verrokkiryhmään:**

**Kommuner som har valts till försöks- eller kontrollgruppen:**

```

# Code author:
# Ramin Izadi
# ramin.izadi@aalto.fi

rm(list=ls())
library(tidyverse)
library(haven)
library(gtools) # for odd
library(rstatix) # for t_test
library(future.apply) # for future_replicate

sessionInfo()
set.seed(195434)

load("final_sample/kunnat_augmented.Rdata")

# Tidy small munis -----
-----

kunnat_pienet = kunnat %>%

```

```

filter(kori == "pieni",
       saamelaisalue == "Ei") %>% # leaves out Inari
mutate(kori = avi,
       vaka_n16_scaled = vaka_n16*kelpoiset_yksikot_osuus)

# Number of potential allocations in each kori
kunnat_pienet %>% group_by(kori) %>% summarise(a = choose(n(),
round(n()*0.5)))

# Simulate assignment vectors and calculate distance -----
--

# Continuous covariates to balance
covariate_vector = c("osuus_ulkomaalaistaustaiset",
                    "osuus_toinen_aste", "osuus_korkea_aste",
                    "tulot_2018avg",
                    "vaka_n16_scaled", "eivaka_n16",
                    "odotettu_kustannus16")

# Calculate covariance matrix for Mahalanobis distance
covariance = kunnat_pienet %>%
  select(all_of(covariate_vector)) %>% cov

# Rerandomization function
mahalanobis_distance = function(assignment_share = 0.5, ...){

  # From each group, randomly choose half into treatment (rounded down)
  koeryhma = kunnat_pienet %>% group_by(kori) %>%
    slice_sample(prop = assignment_share) %>% # slice_sample rounds down
    pull(kunta)

  # Choose an extra treatment muni from half of the odd groups
  misfits = kunnat_pienet %>% group_by(kori) %>%
    filter(odd(n())) %>% # groups with odd number of munis
    filter(!kunta %in% koeryhma) %>% # available munis (not yet chosen
for treatment)
    slice_sample(n = 1) %>% ungroup %>% # choose one muni from each odd
group
    slice_sample(prop = assignment_share) %>% # choose half into
treatment group
    pull(kunta)

  koeryhma = c(koeryhma, misfits)

  verrokkiryhma = kunnat_pienet %>% filter(!kunta %in% koeryhma) %>%
pull(kunta)

  # Mean of the covariates in the treatment group
  treat = kunnat_pienet %>%
    filter(kunta %in% koeryhma) %>%
    select(...) %>%
    summarise_all(mean) %>% as.matrix

  # Mean of the covariates in the control group
  control = kunnat_pienet %>%

```

```

filter(kunta %in% verrokkiryhma) %>%
select(...) %>%
summarise_all(mean) %>% as.matrix

# Mahalanobis distance between the mean vectors
# (Euclidian distance between the two means in a transformed space)
distance = mahalnobis(x = treat, center = control, cov = covariance)
%>% sqrt

# Save into a tibble
tibble(distance = distance, koeryhma = list(koeryhma), verrokkiryhma =
list(verrokkiryhma))
}

# Generate admissible allocations
# Number of replications: 1 000 000 (11h with parallel processing)

plan(multisession)
system.time(
  # Parallel processing function that replicates the above function
  (assignment_vectors = future_replicate(1000000,
mahalanobis_distance(assignment_share = 0.5,
all_of(covariate_vector)),
simplify = F,
future.seed = TRUE))
)

assignment_vectors = bind_rows(assignment_vectors)
assignment_vectors

# Choose the most balanced subset and pick the final treatment assignment
vector -----
-----

# Choose restricted assignment set H and add mirror allocations
# If Fisher exact p-value < 0.01 then H > 200

H = assignment_vectors %>% arrange(distance) %>% filter(row_number() %in%
1:1000)
H_mirror = H %>% rename(koeryhma = verrokkiryhma, verrokkiryhma =
koeryhma)
H = bind_rows(H, H_mirror)

# Save the subset of admissible allocations (Use this for exact p-value
calculations)
save(H, file = "final_sample/assignment_vectors_subset.Rdata")

# Choose final treatment assignment vector and save it
koekunnat = H %>% sample_n(1) %>%
gather(key = treat, value = kunta, koeryhma, verrokkiryhma) %>%
unnest(cols = kunta)

kunnat_pienet_satunnaistaminen_lopullinen = koekunnat %>%
left_join(select(kunnat, nimi, kunta, avi, kieli)) %>% arrange(avi) %>%
select(nimi, kunta, treat, avi, kieli)

```

```
write_excel_csv2(kunnat_pienet_satunnaistaminen_lopullinen,
"final_sample/pienten_kuntien_arvontatulos.csv")

# Code author:
# Ramin Izadi
# ramin.izadi@aalto.fi

rm(list=ls())
library(tidyverse)
library(haven)
library(gtools) # for odd
library(rstatix) # for t_test
library(future.apply) # for future_replicate

sessionInfo()
set.seed(132345)

load("final_sample/kunnat_augmented.Rdata")

saamelaisalueet_satunnaistaminen_lopullinen = kunnat %>%
filter(saamelaisalue != "Ei", yksikot > 0) %>%
  mutate(treat = ifelse(nimi == sample(nimi, 1), "koeryhma",
"verrokkiryhma")) %>%
  select(nimi, kunta, treat, avi, kieli)

write_excel_csv2(saamelaisalueet_satunnaistaminen_lopullinen,
"final_sample/saamelaisalueet_arvontatulos.csv")
```