

**Dragehode**

# Table of contents

<b>Velkommen</b>	<b>3</b>
<b>1 Introduksjon</b>	<b>4</b>
1.1 Finansiering . . . . .	4
<b>2 Lokale estimater av populasjonsstørrelse</b>	<b>5</b>
<b>3 Lokale estimater av populasjonsvekstrate</b>	<b>18</b>
<b>4 Vektete populasjonsvekstrater</b>	<b>31</b>
<b>5 Direction plots</b>	<b>33</b>
<b>6 Code to recreate the analysis</b>	<b>61</b>
<b>7 Funksjoner</b>	<b>70</b>

# Velkommen

Dette er nettstedet for overvåkingsrapporten for Dragehode i 2024.

Denne nettsiden er og vil alltid være gratis, lisensiert under CC BY-NC-ND 3.0-lisensen.

# 1 Introduksjon

Denne nettsiden presenterer resultater fra overvåking av dragehode *Dracocephalum ruysciana* i Norge. Dragehode er kategorisert som sårbar (VU) på den norske rødlista for arter, og dragehode er en prioritert art, med egen forskrift og handlingsplan.

Du kan lese mer om dragehode og overvåkingsprosjektet på [NINAs nettsider](#)

Årsrapporten for 2022, som denne nettsiden refererer til, finner du [her](#)

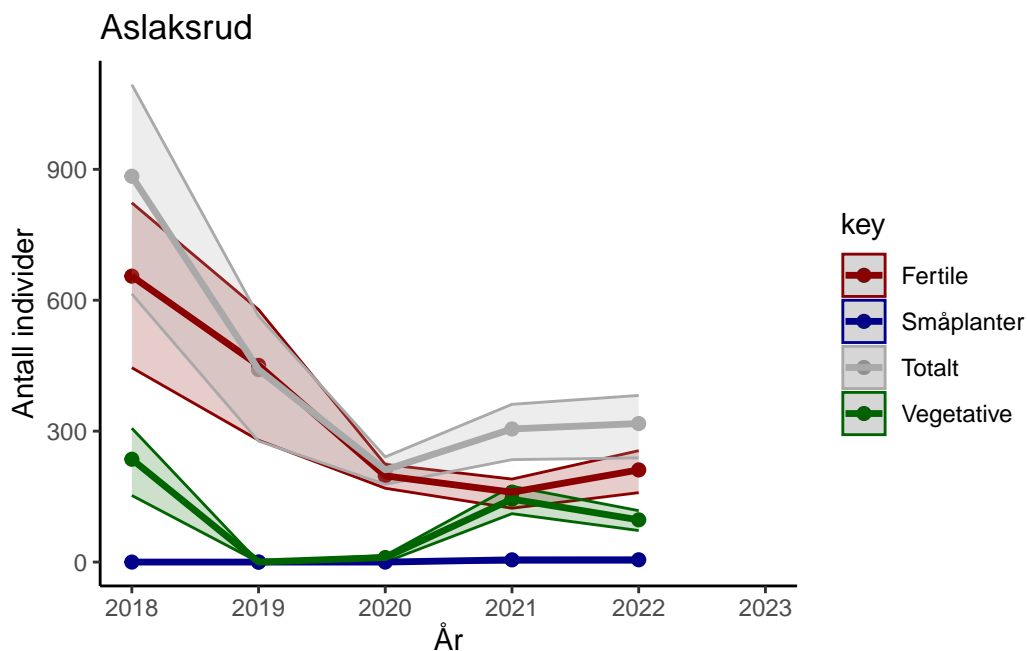
## 1.1 Finansiering

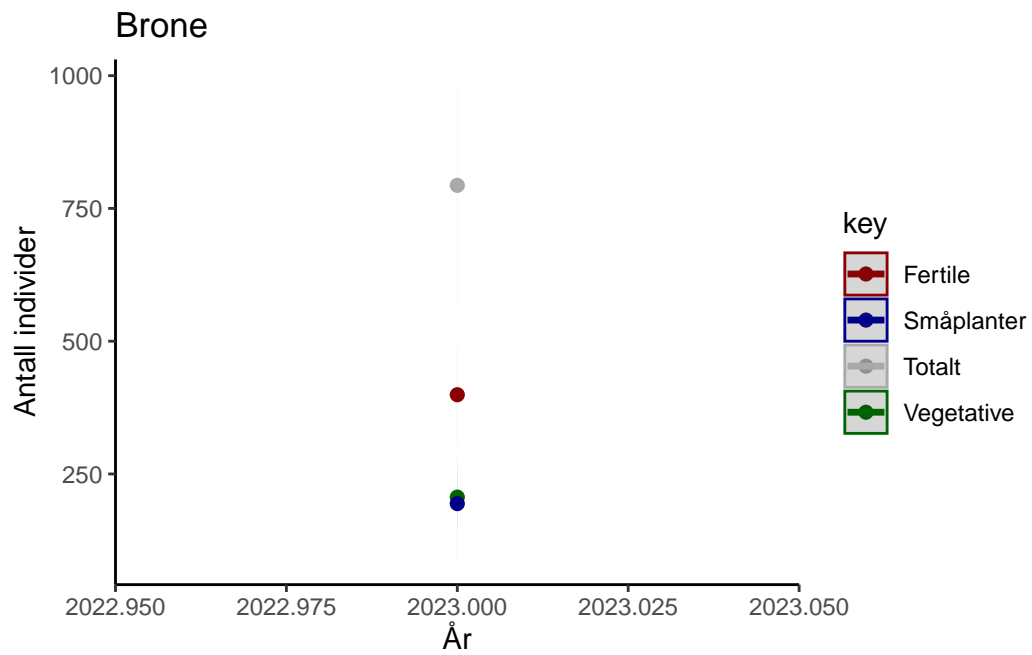
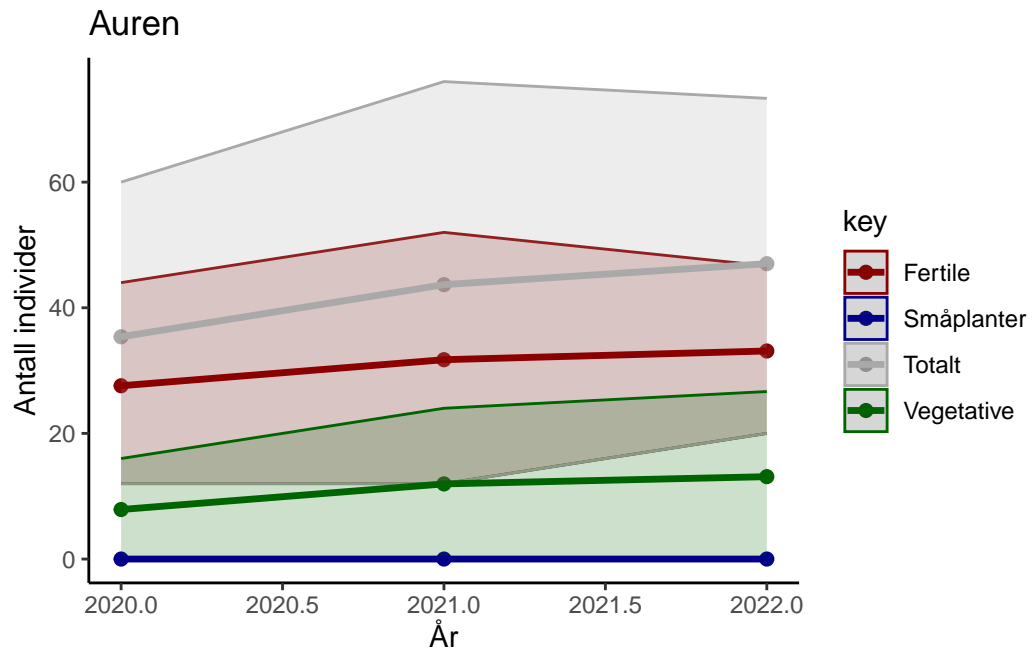
Overvåking av dragehode finansieres av Statsforvalteren i Oslo og Viken.

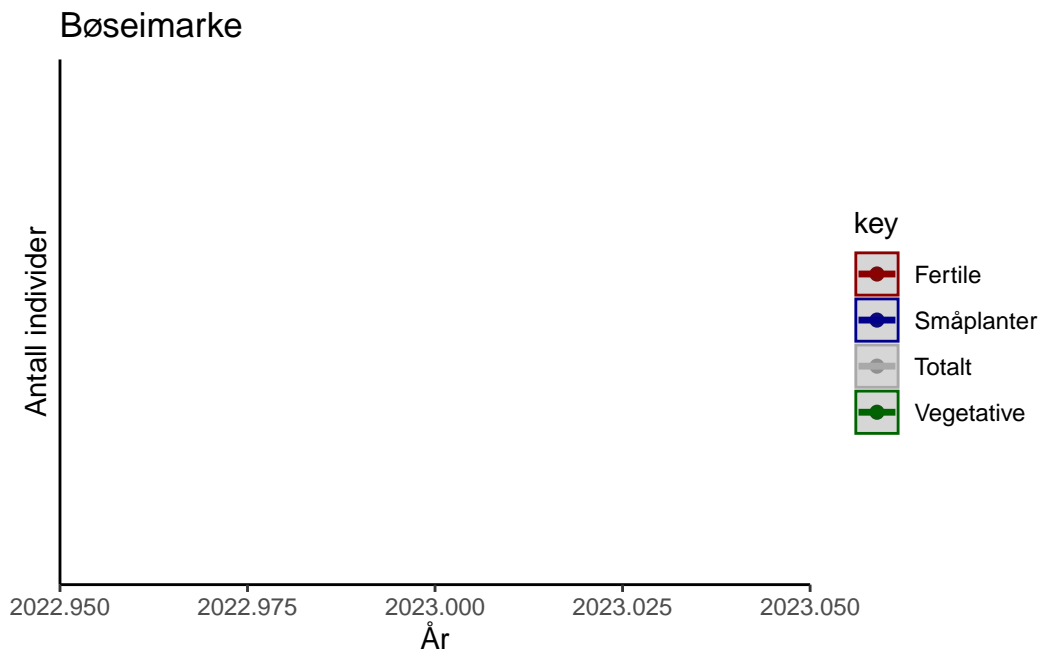
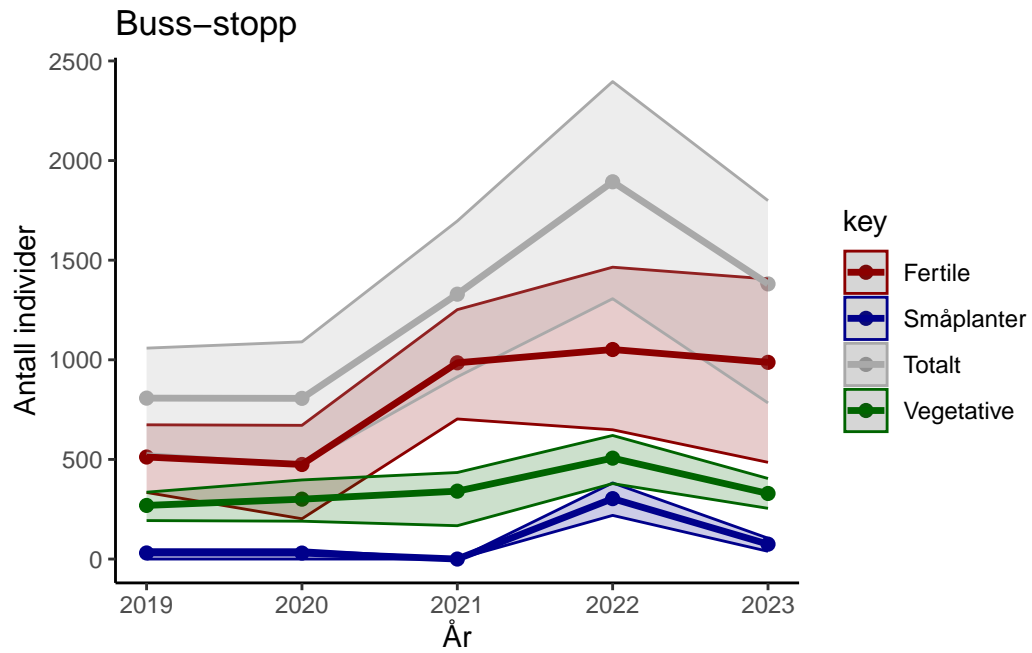
## 2 Lokale estimater av populasjonsstørrelse

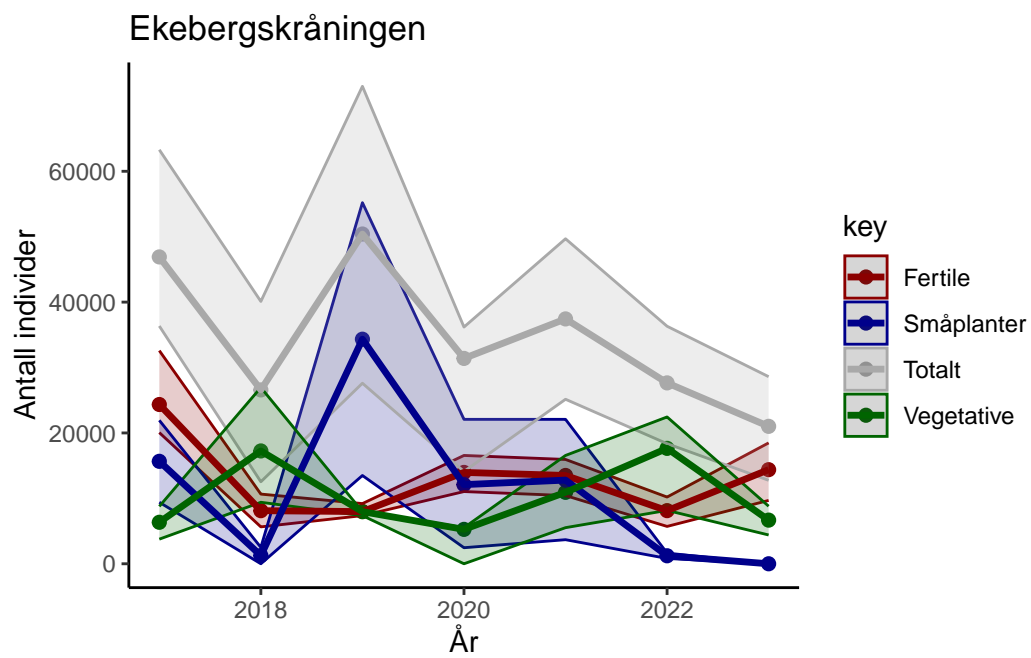
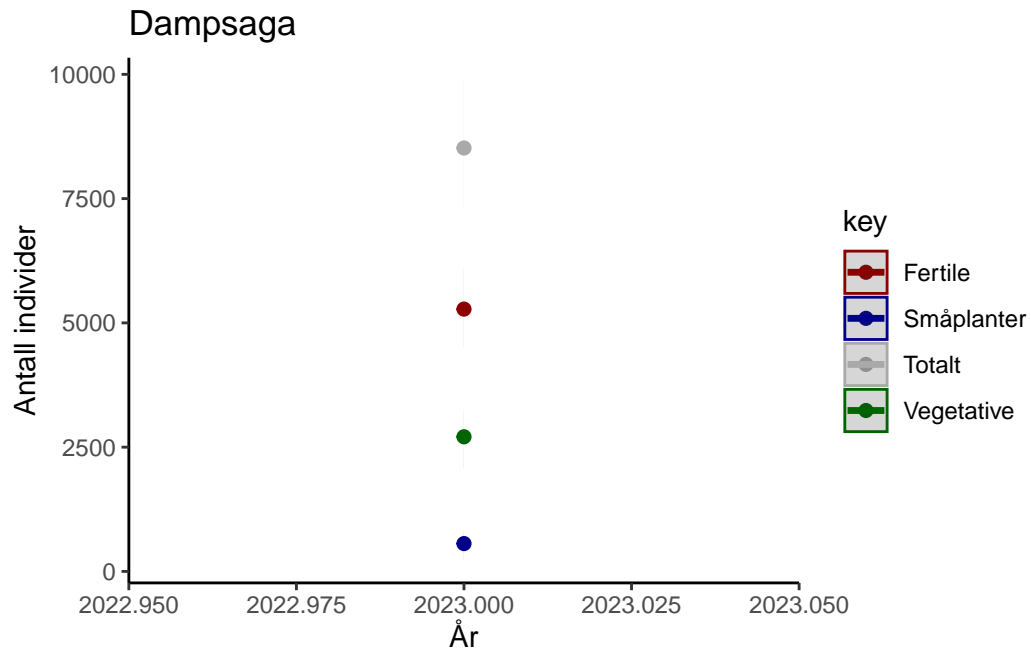
Disse figurene viser estimert populasjonsstørrelse, totalt og fordelt på de tre størrelsesklassene fertile, vegetative og småplanter, for hvert år i de 25 lokalitetene som inngår i overvåkingen.

Populasjonsstørrelse er estimert ved å bruke tettheten av individer i overvåkingsrutene, samt arealet dragehode forekommer på (registrert i forekomstruter langs transekter) innenfor lokaliteten. Usikkerheten i populasjonsestimatene ble beregnet ved tilfeldige trekninger av forekomstruter og overvåkingsruter ('bootstrapping') med 2000 gjentak. Bootstrapping er en re-samplingsmetode som gjør det mulig å beregne usikkerhet rundt en gjennomsnittsverdi. Strekene viser gjennomsnittsverdi, mens de skraverte feltene viser 95 % konfidensintervaller.

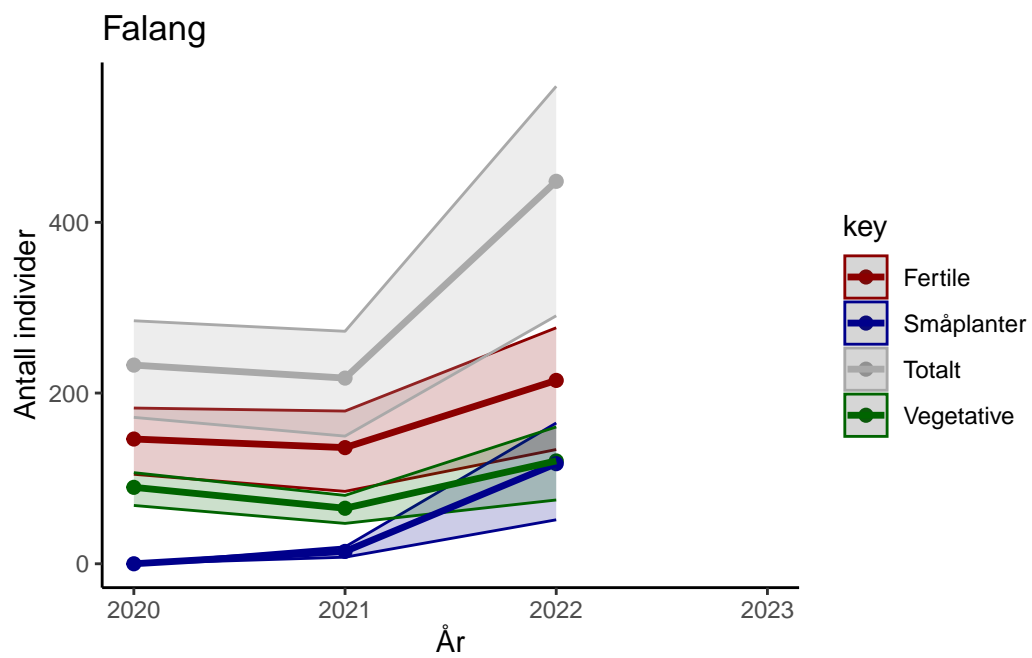
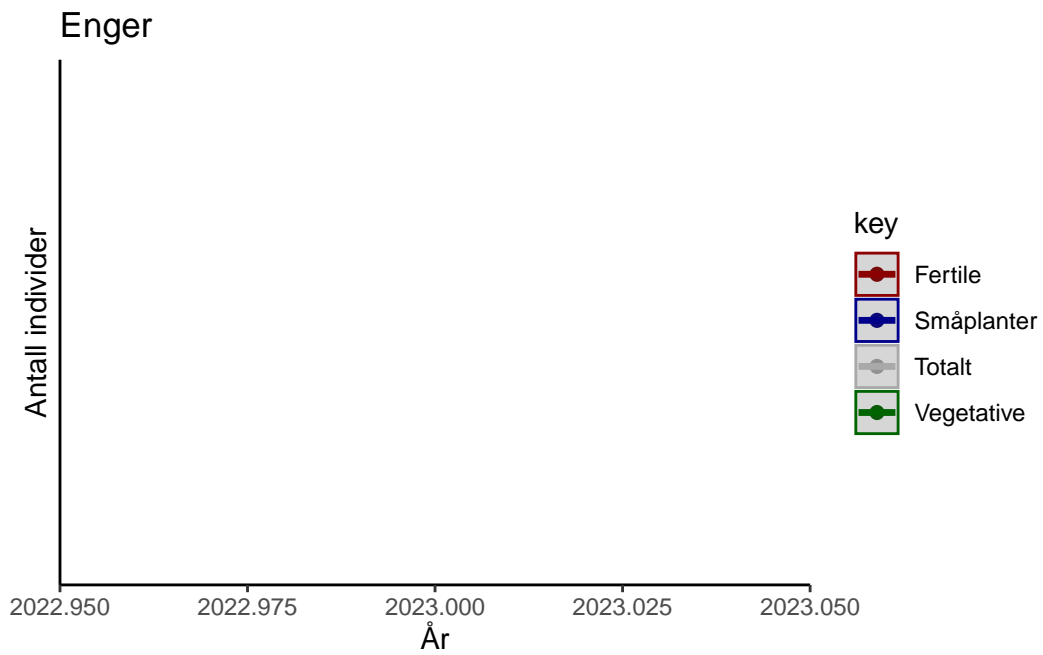


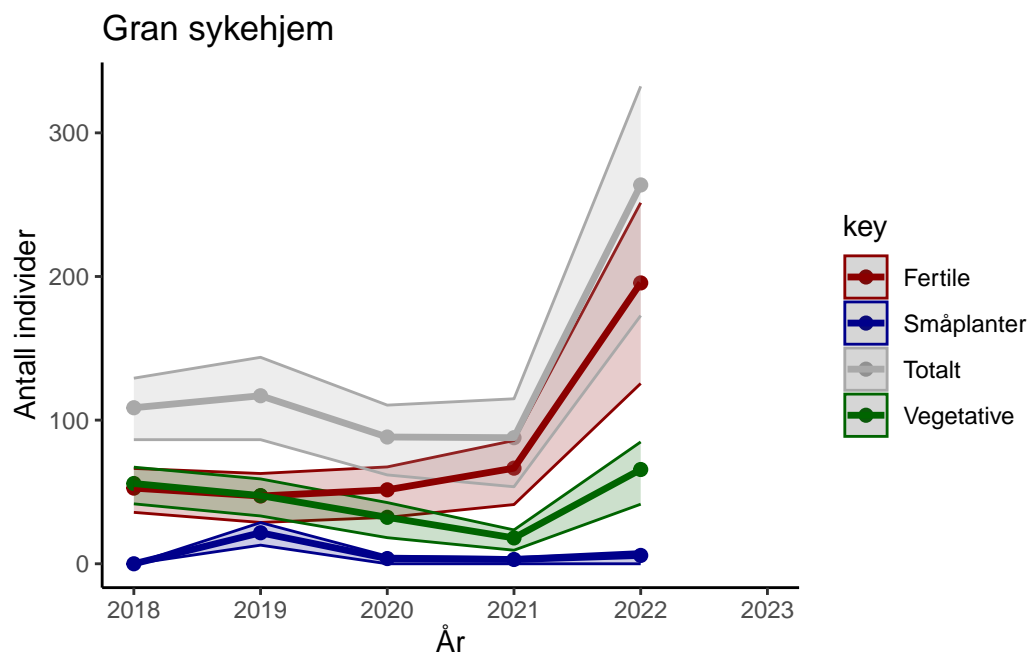
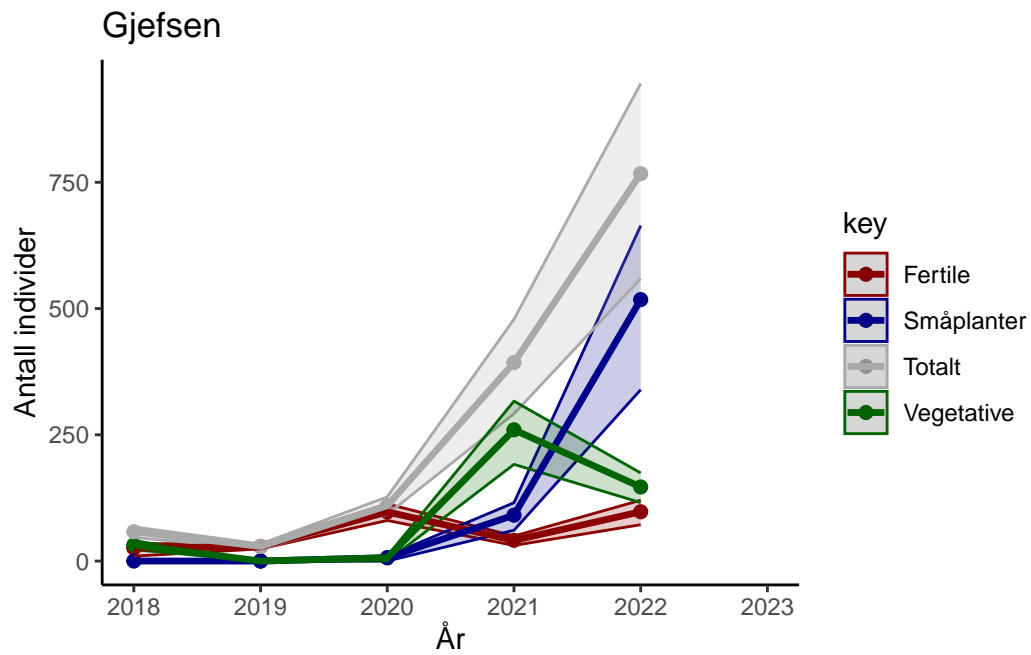


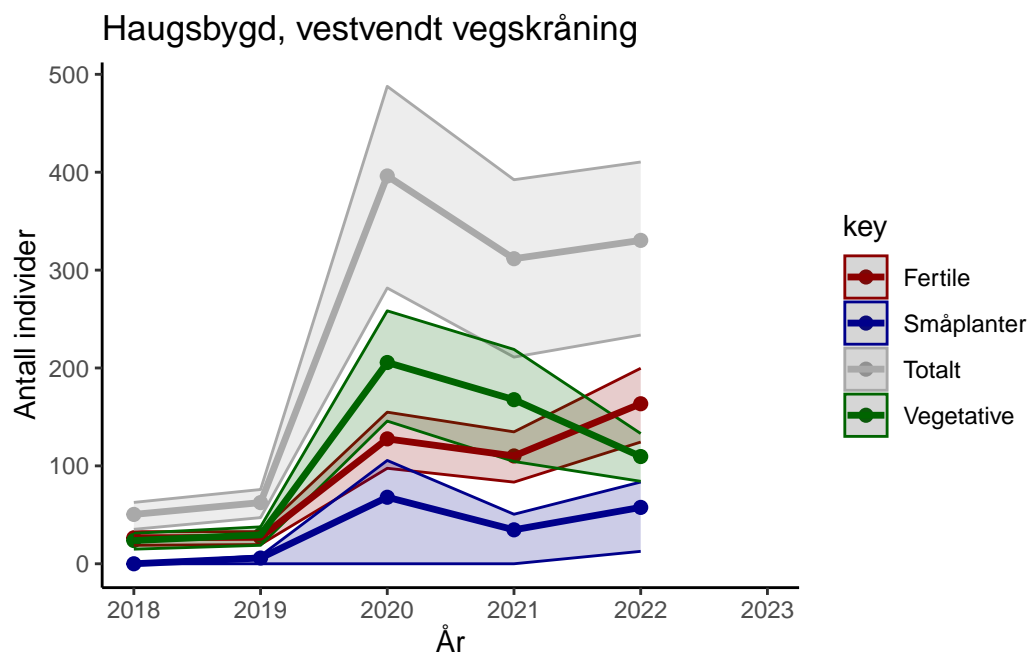
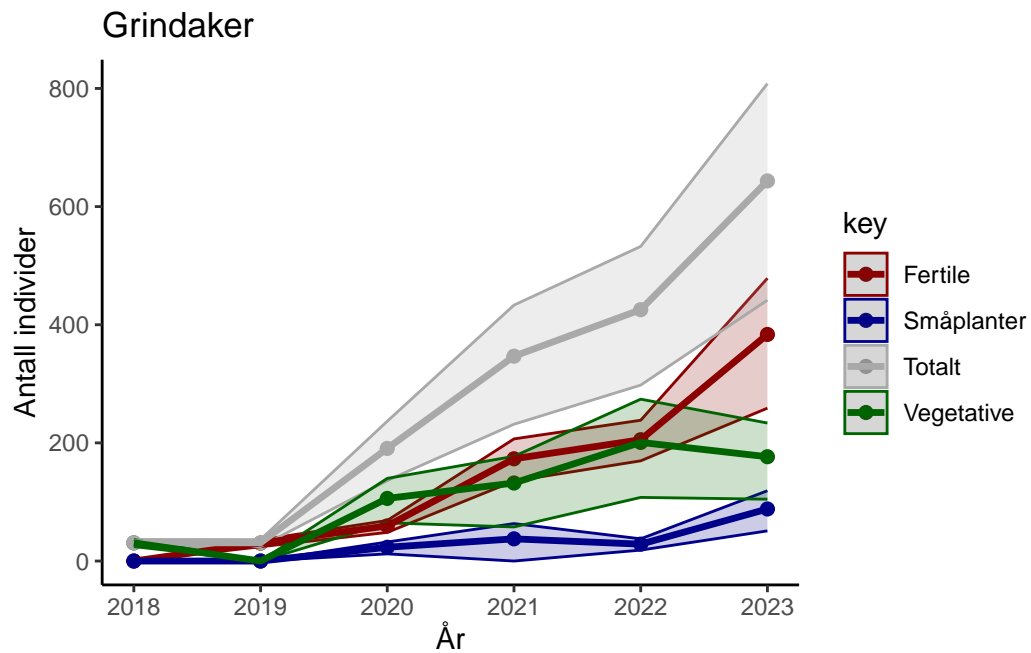


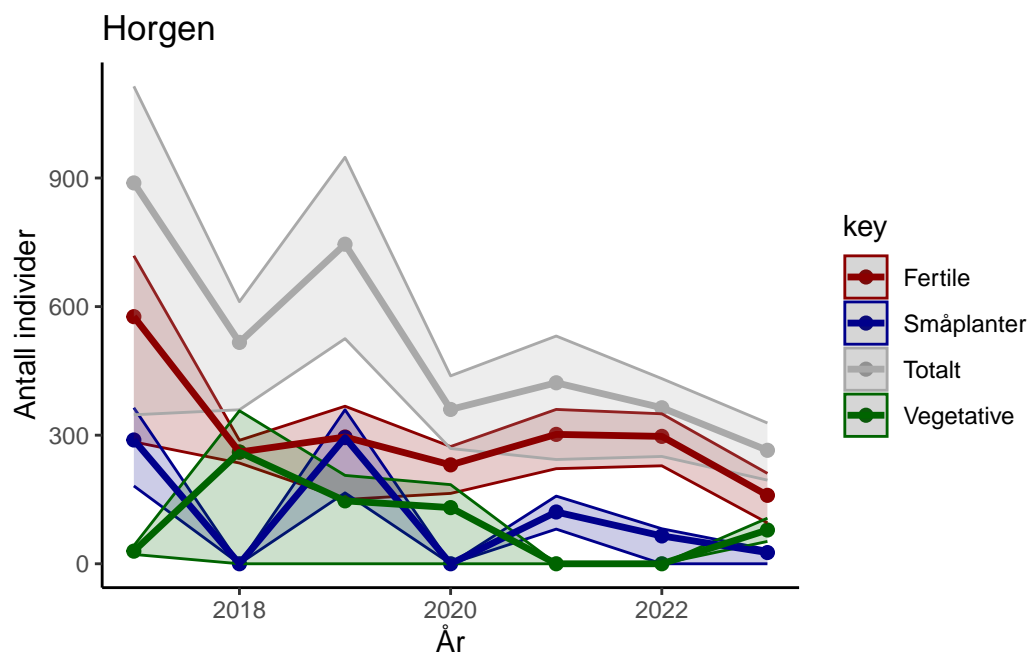
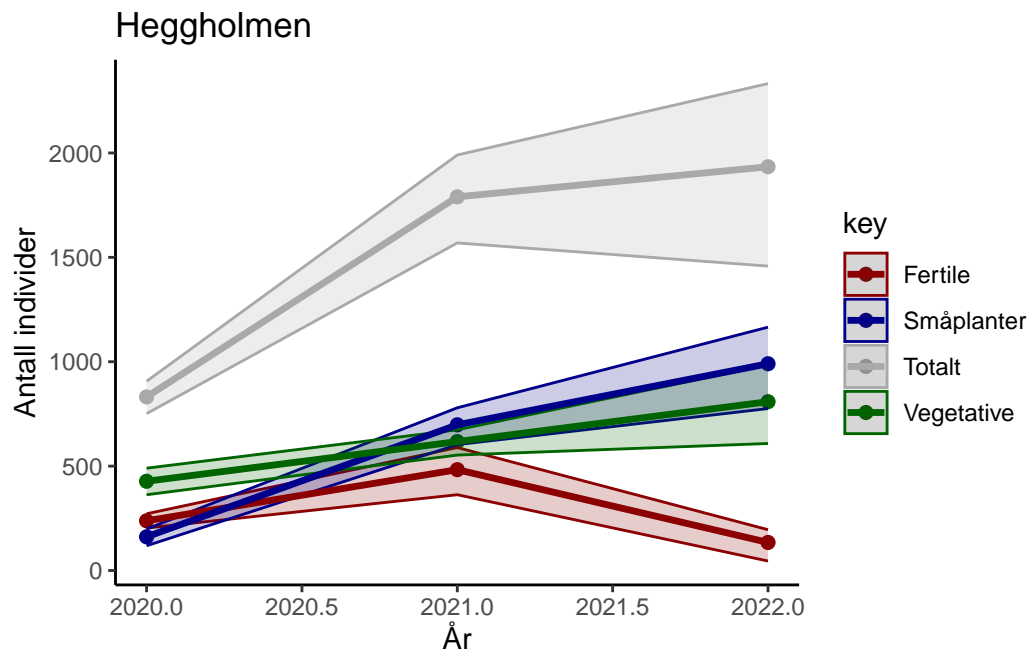


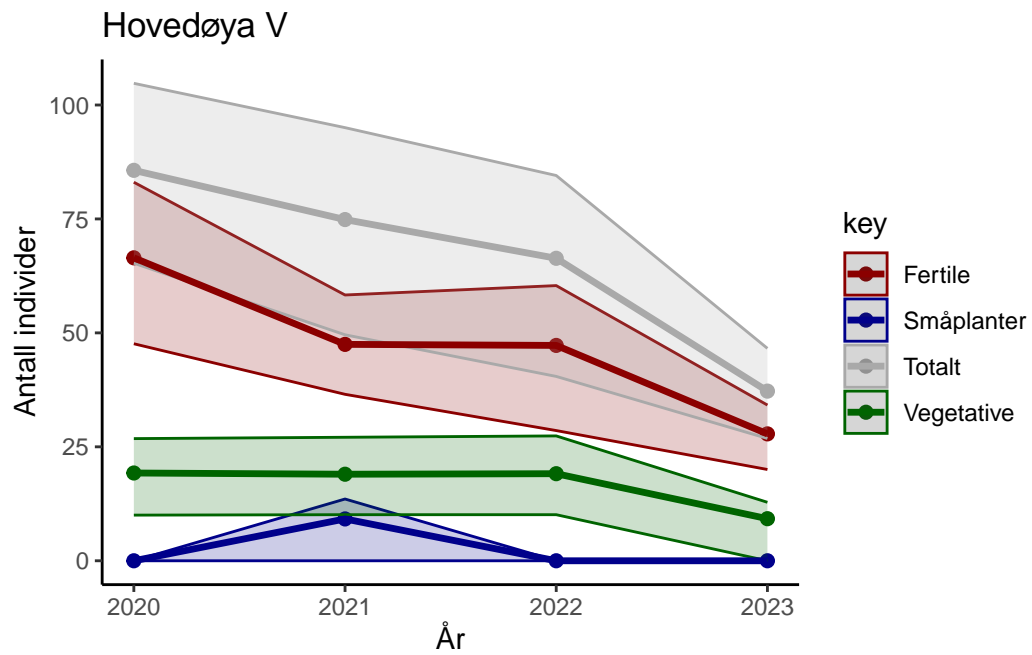
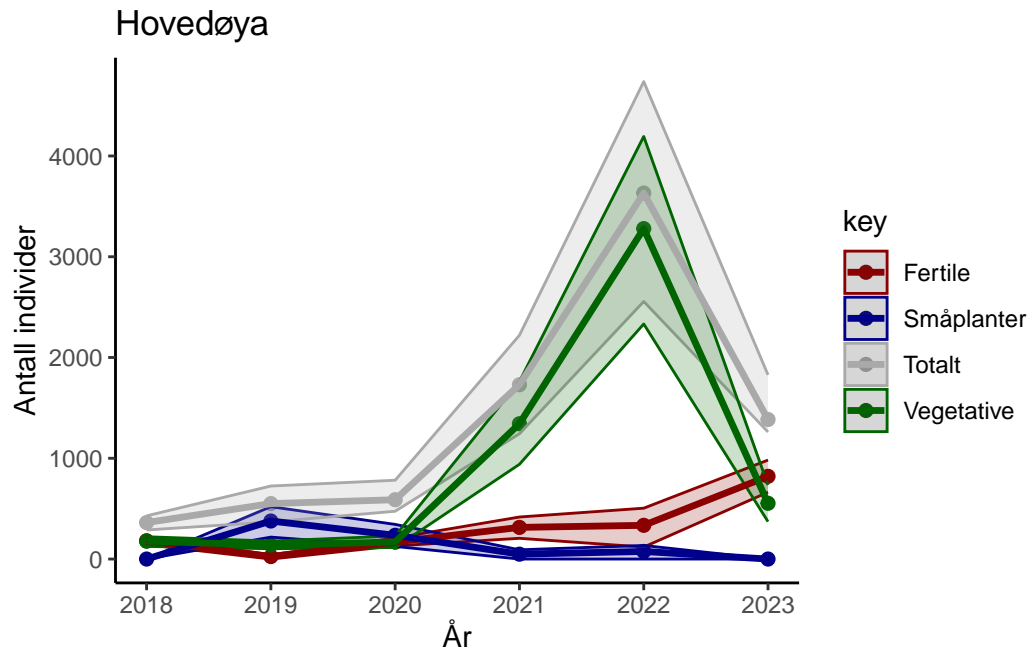


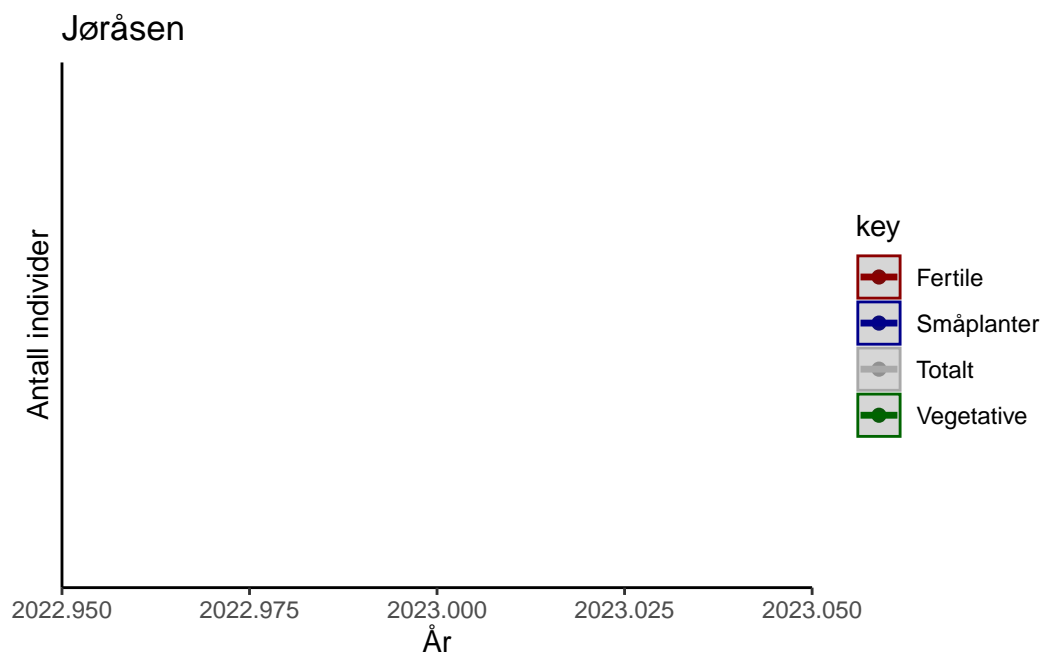
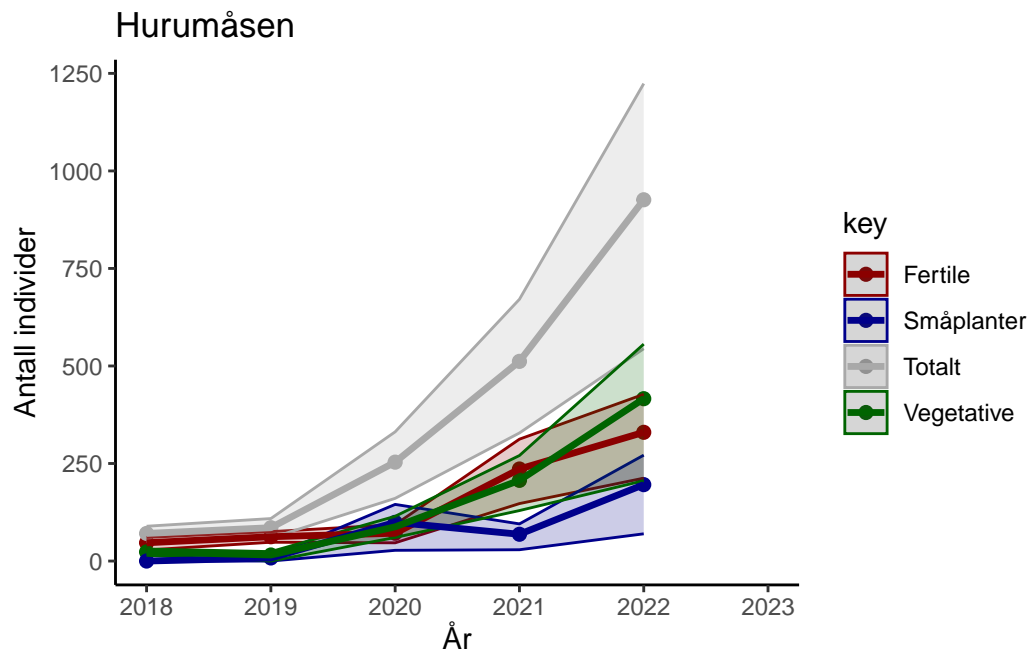


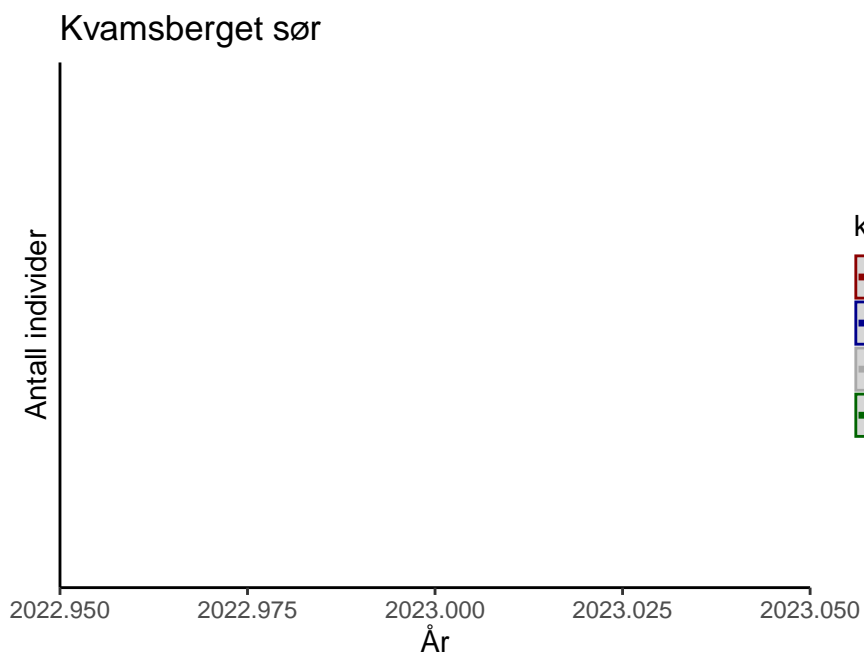
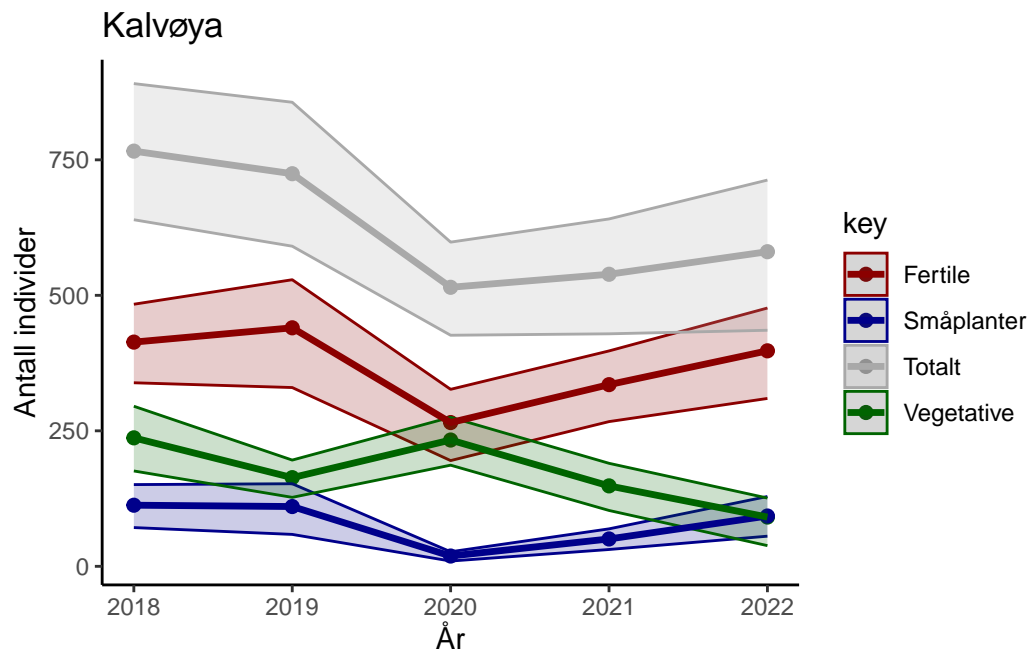


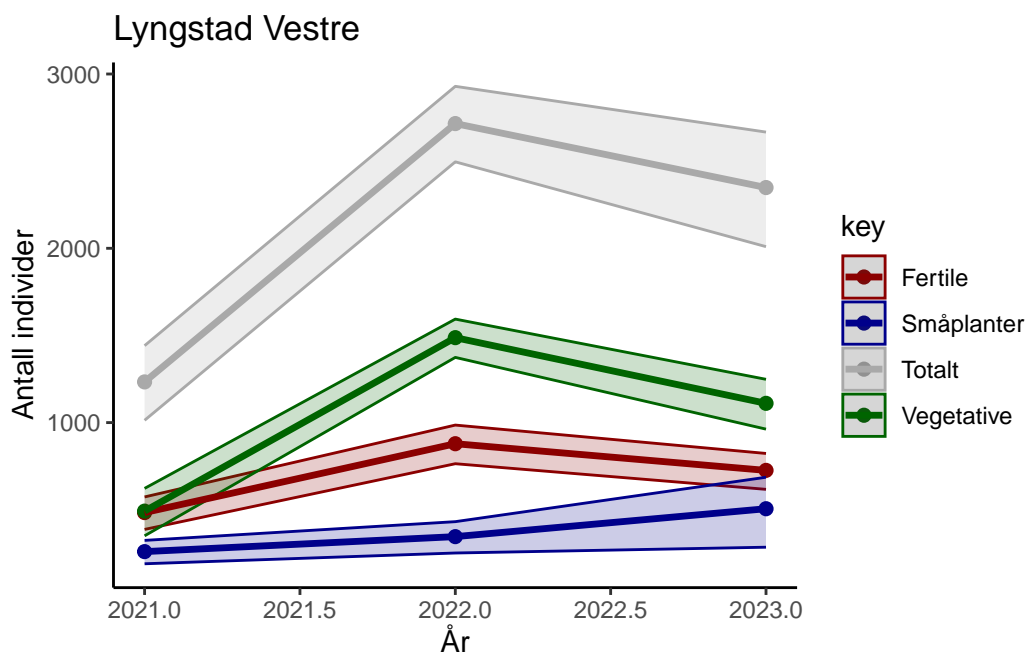
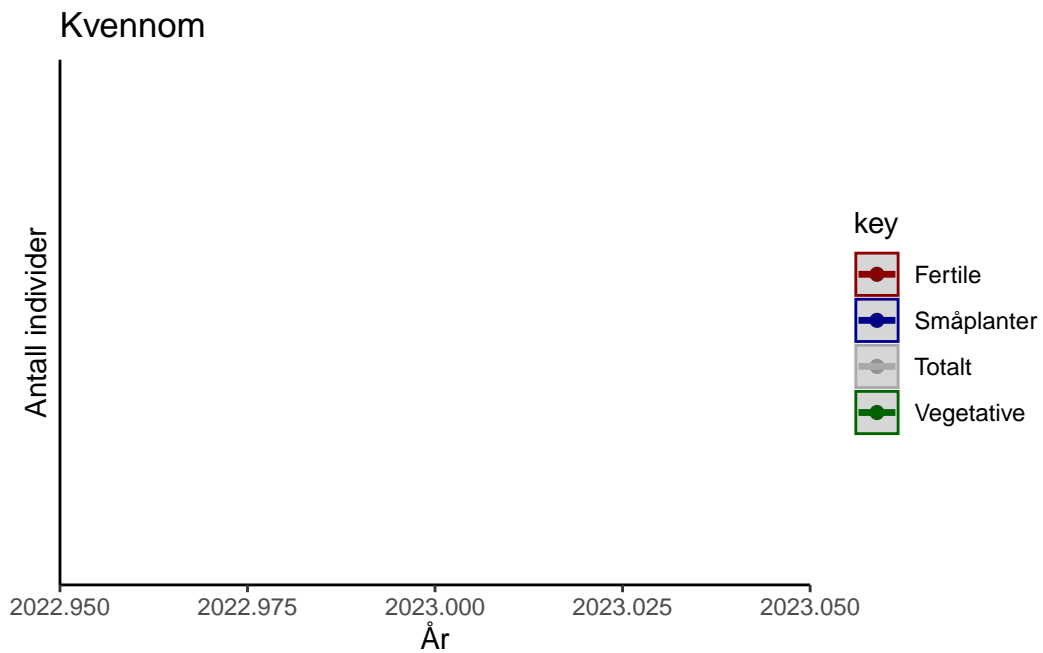




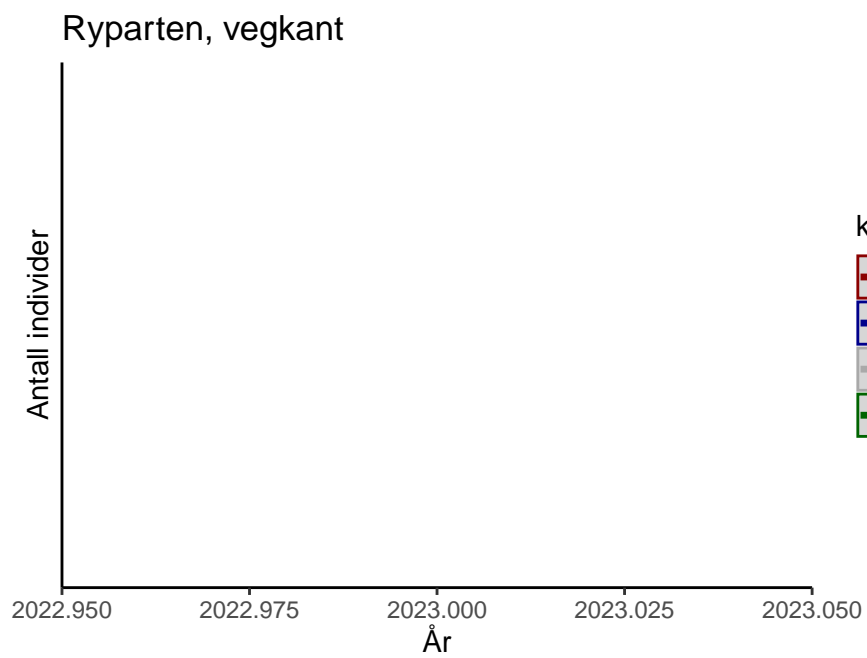
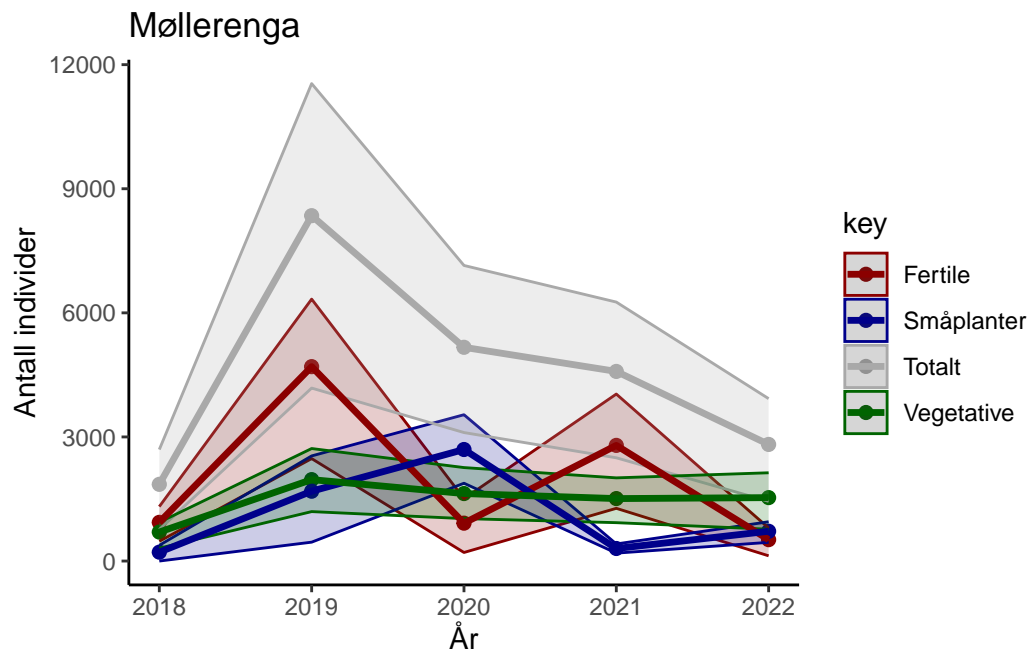










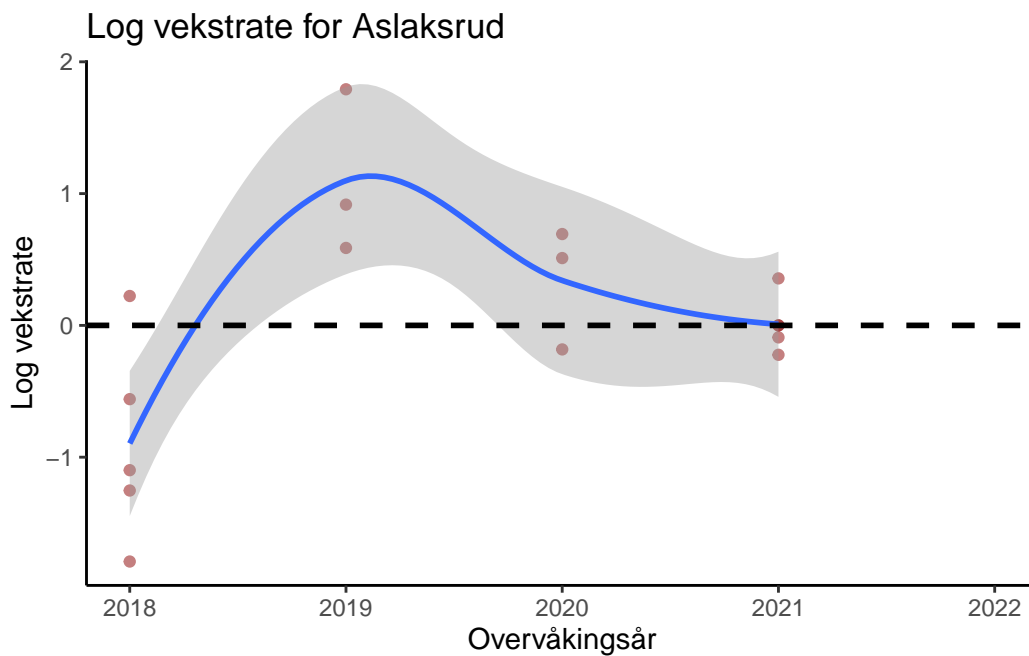


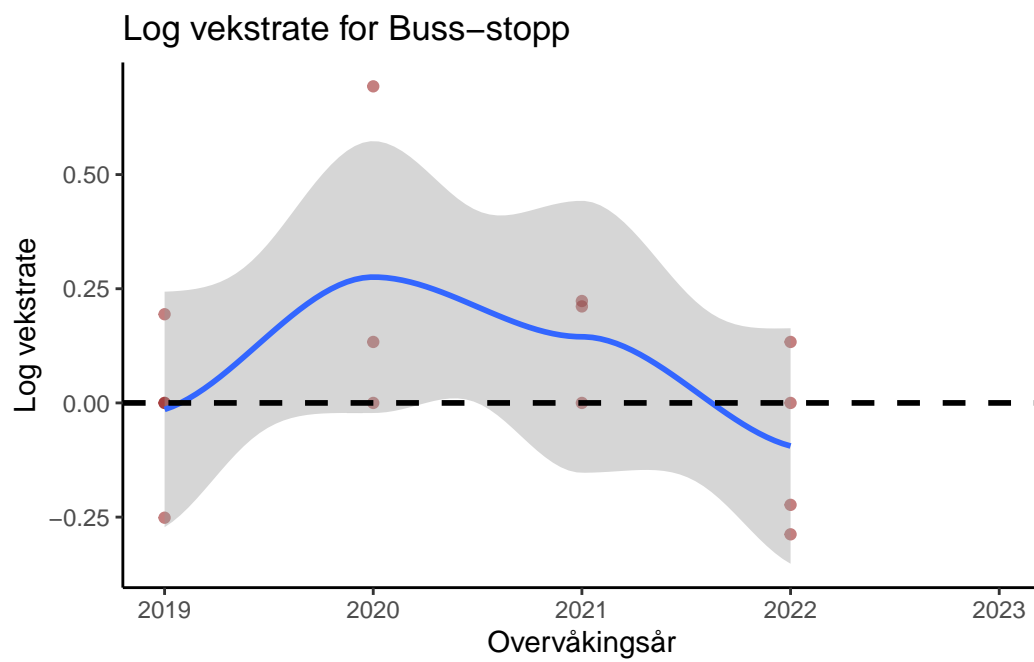
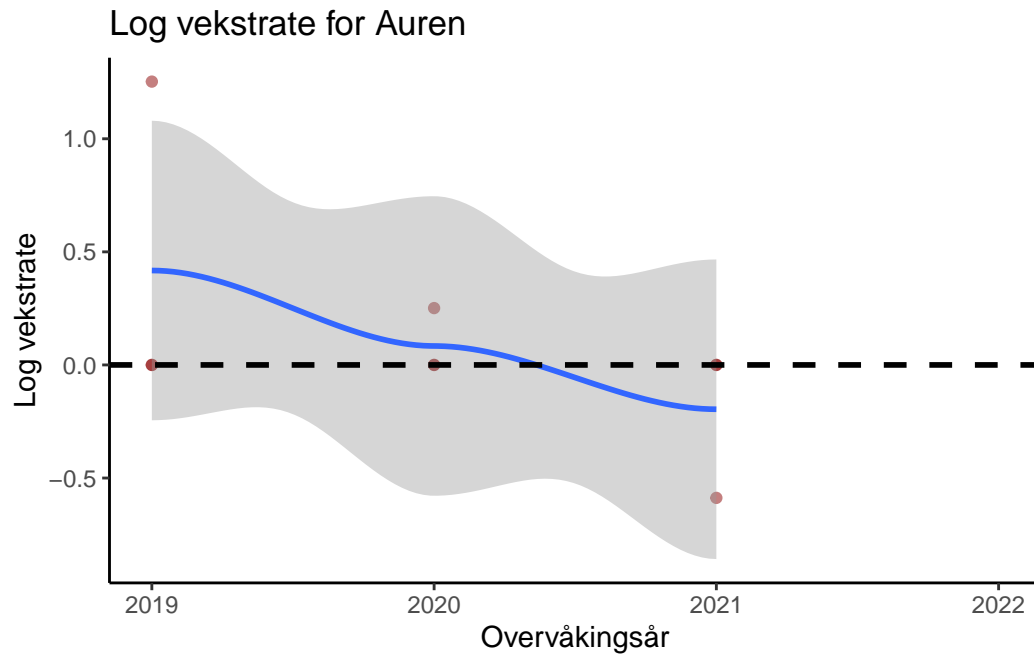
### 3 Lokale estimater av populasjonsvekstrate

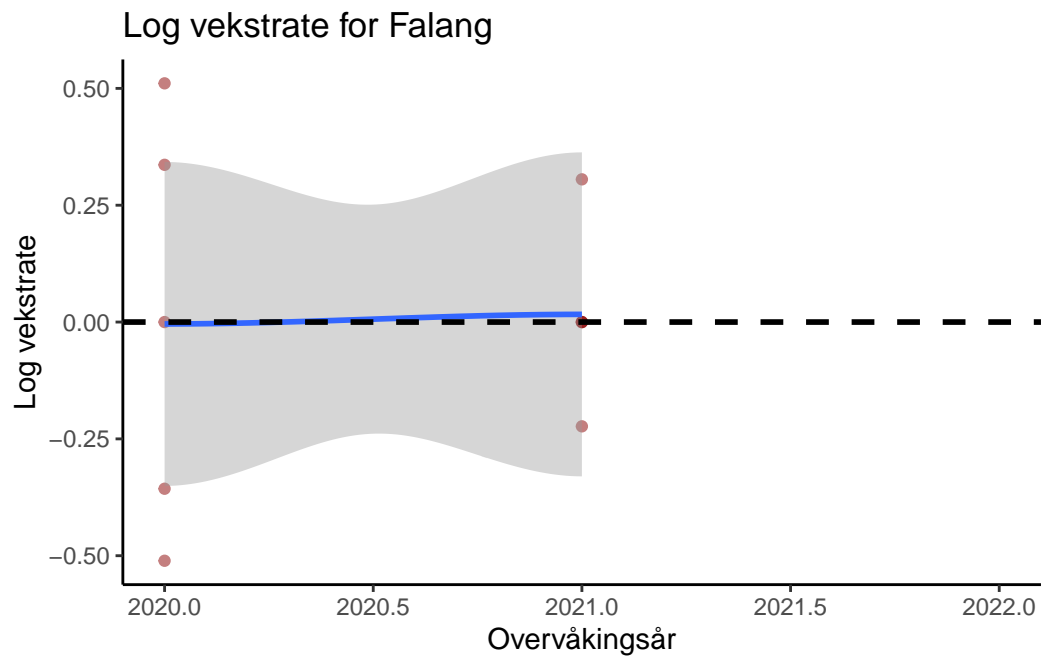
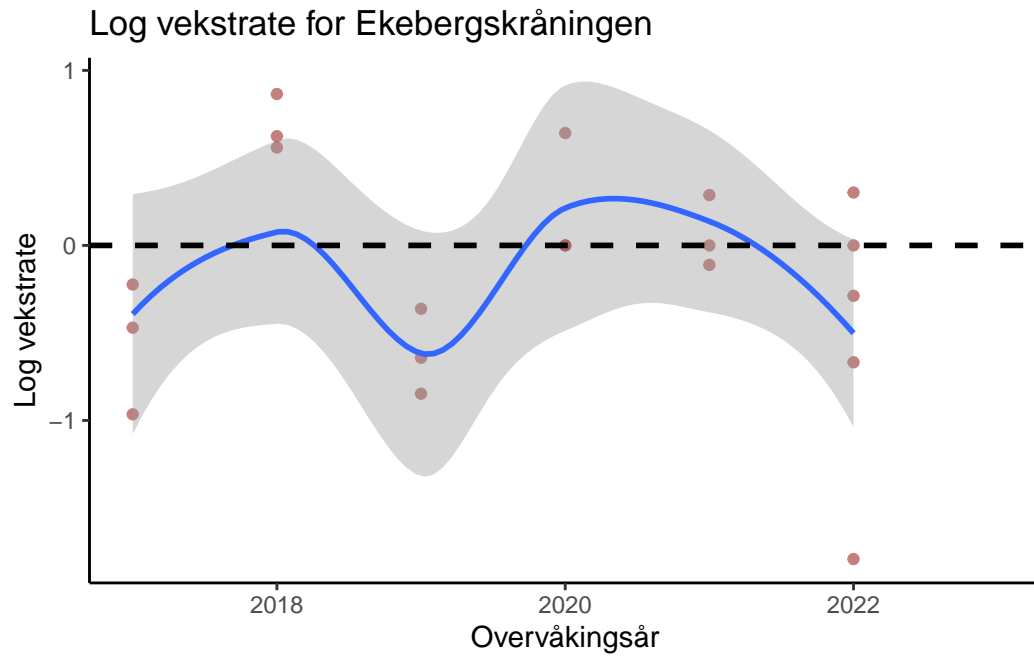
Disse figurene viser estimert populasjonsvekstrate for hvert år i de 25 lokalitetene som inngår i overvåkingen.

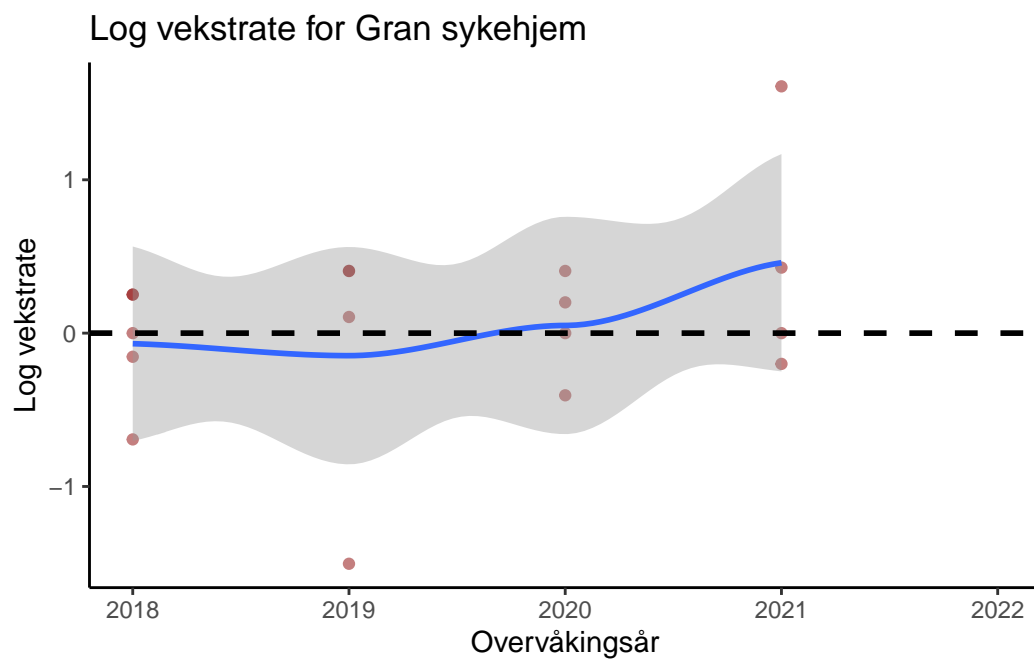
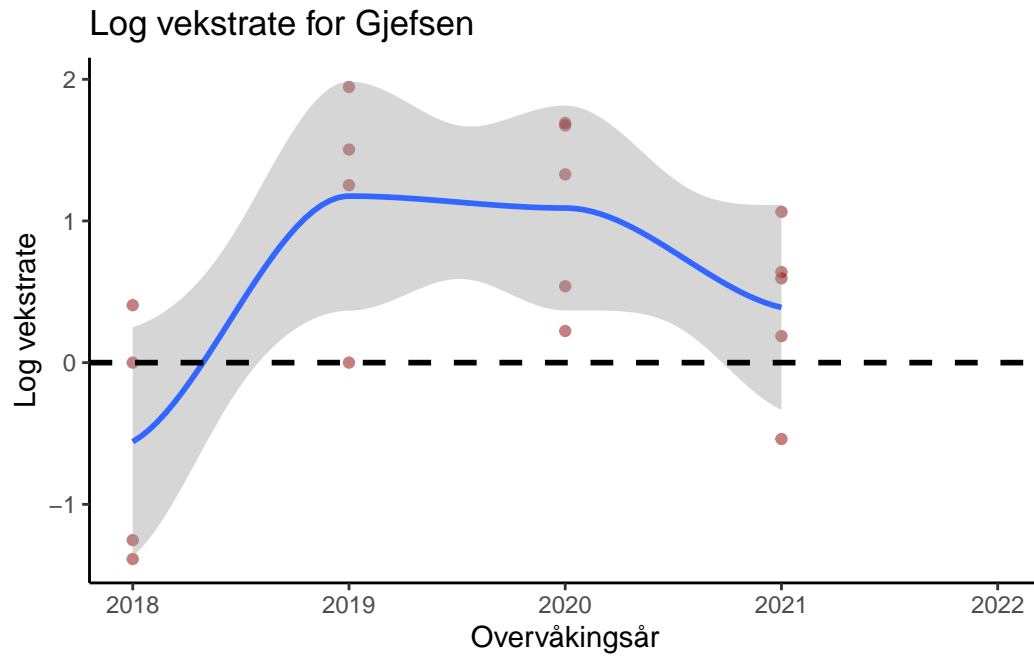
Vekstrate ble beregnet som forholdet mellom populasjonsstørrelsen (antall individer totalt) i år  $t$  over antall individer i år  $t - 1$ . Et gjennomsnitt ble beregnet for hver lokalitet og år, over alle overvåkingsruter på lokaliteten. I figurene er vekstraten log-transformert. En log-vekstrate på 0 betyr at populasjonen er stabil (antall individer i år  $t =$  antall individer i år  $t - 1$ ). En positiv log-vekstrate ( $> 0$ ) betyr en populasjon i økning, mens en negativ log-vekstrate ( $< 0$ ) betyr en populasjon i nedgang. Året langs x-aksen angir år  $t - 1$ , dvs. vekstraten i 2020 representerer endringen som er skjedd fra 2020 til 2021 (log(antall individer i 2021/antall individer i 2020)).

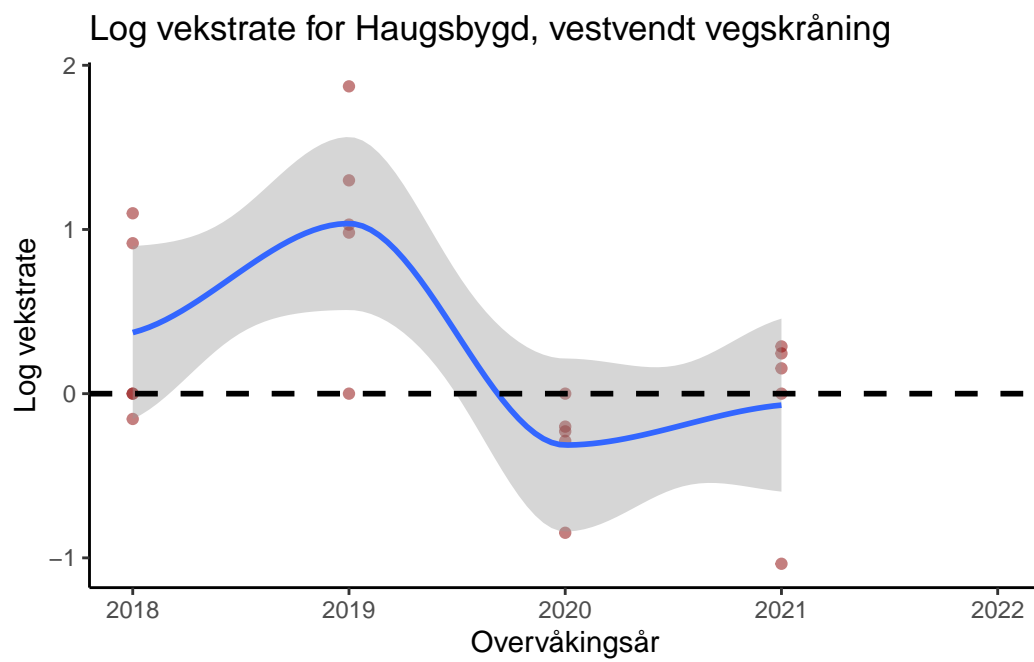
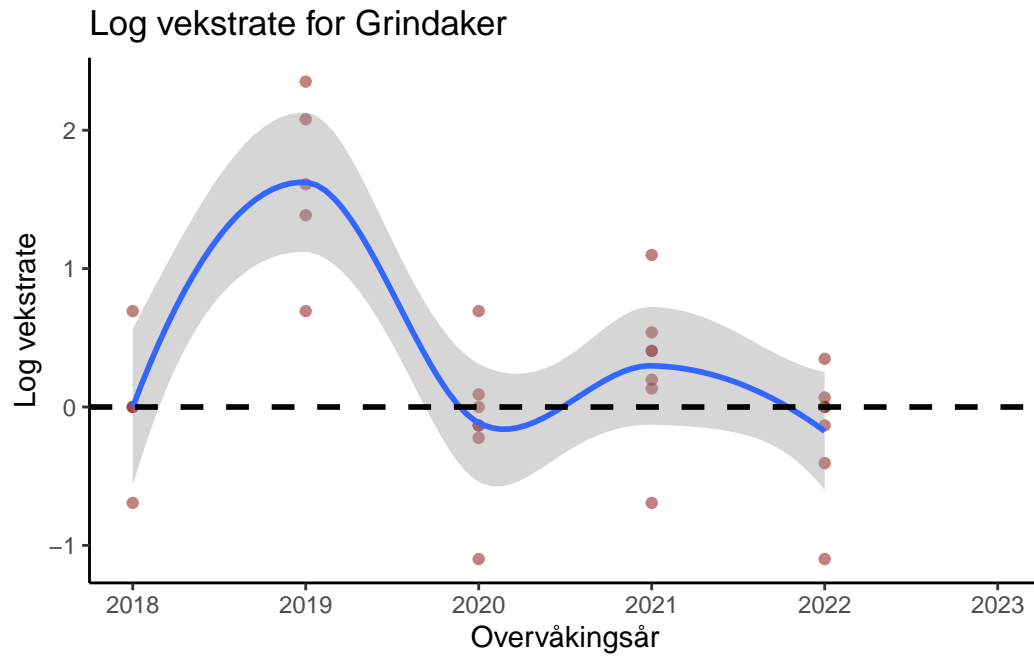
I figurene vises hver overvåkingsrute som en prikk, gjennomsnittsverdien for lokaliteten som tykk blå strek og 95 % konfidensintervall som grått skravert område.

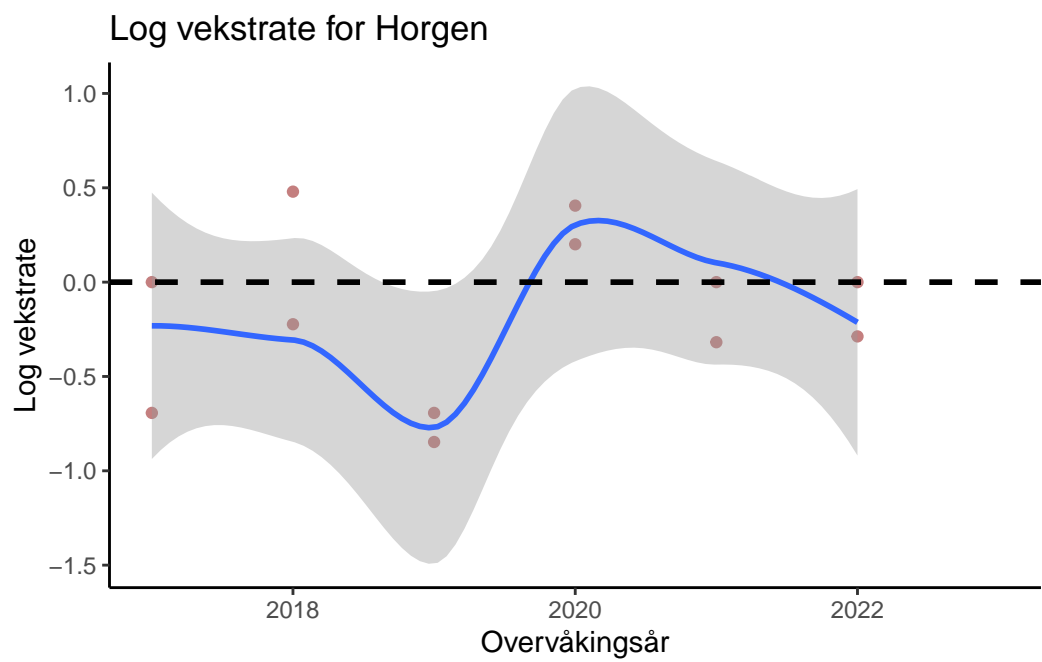
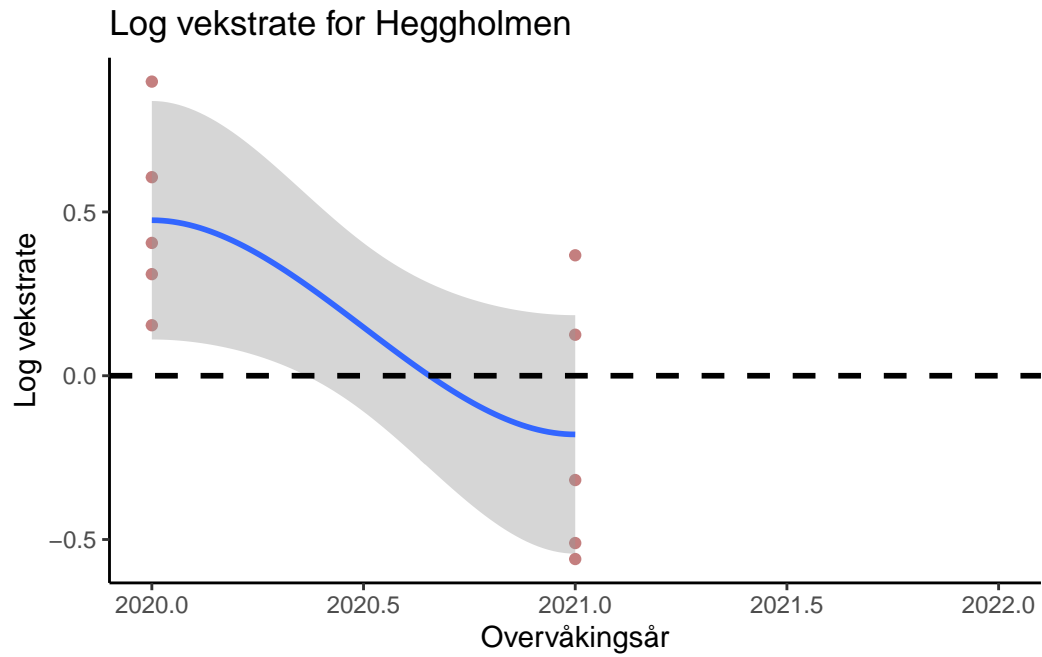


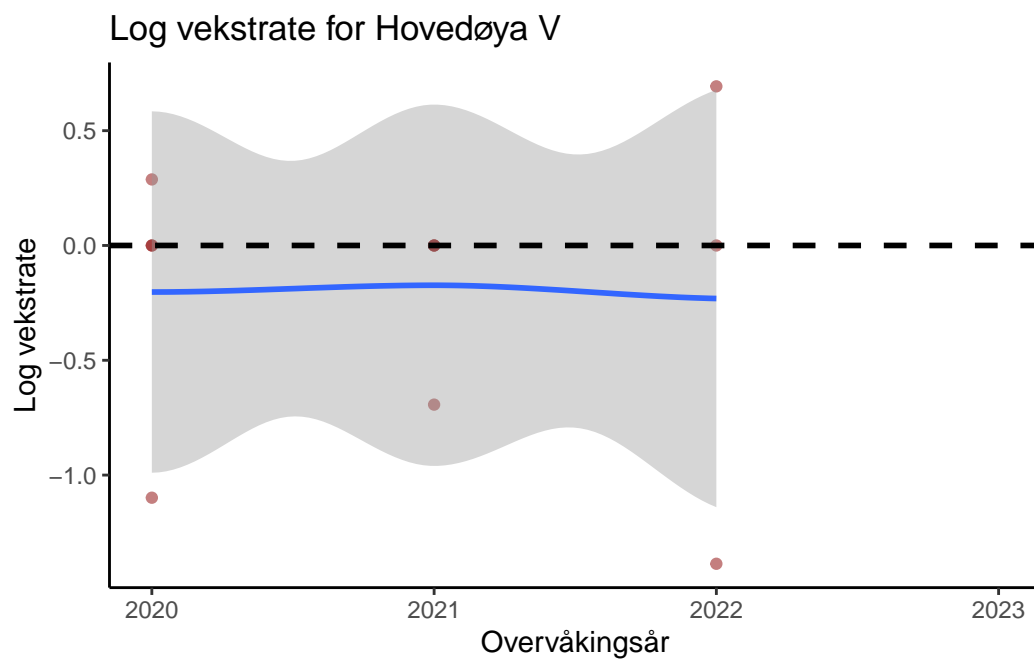
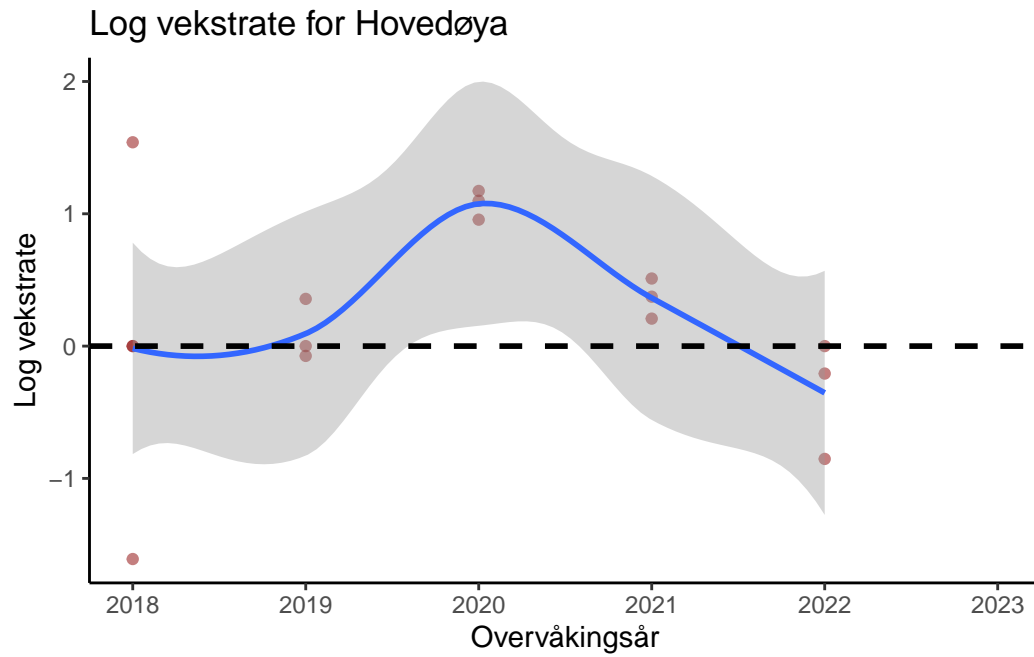




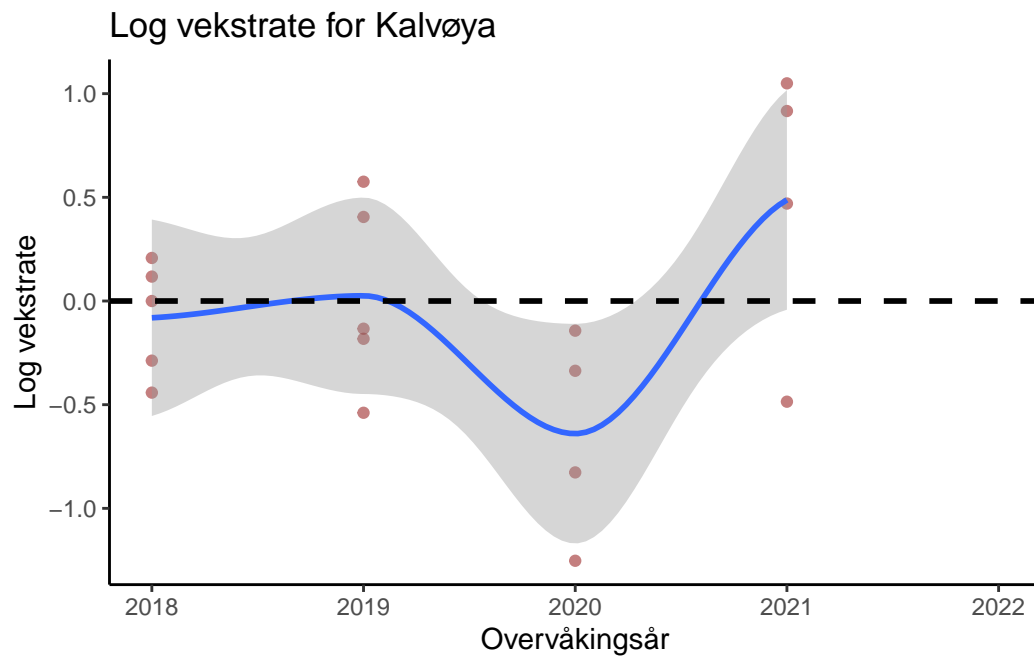
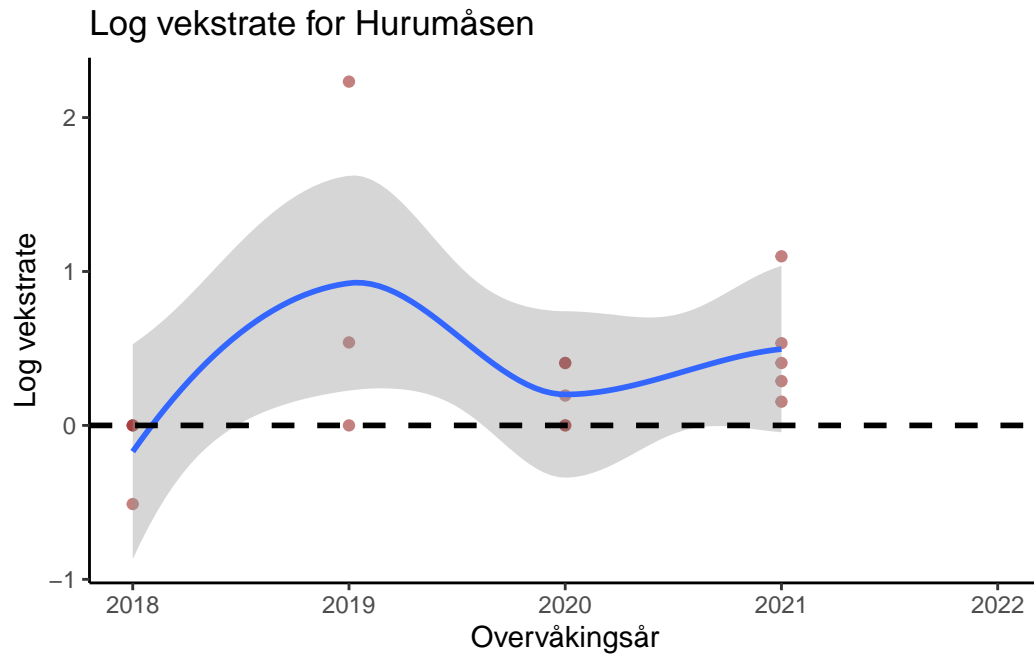


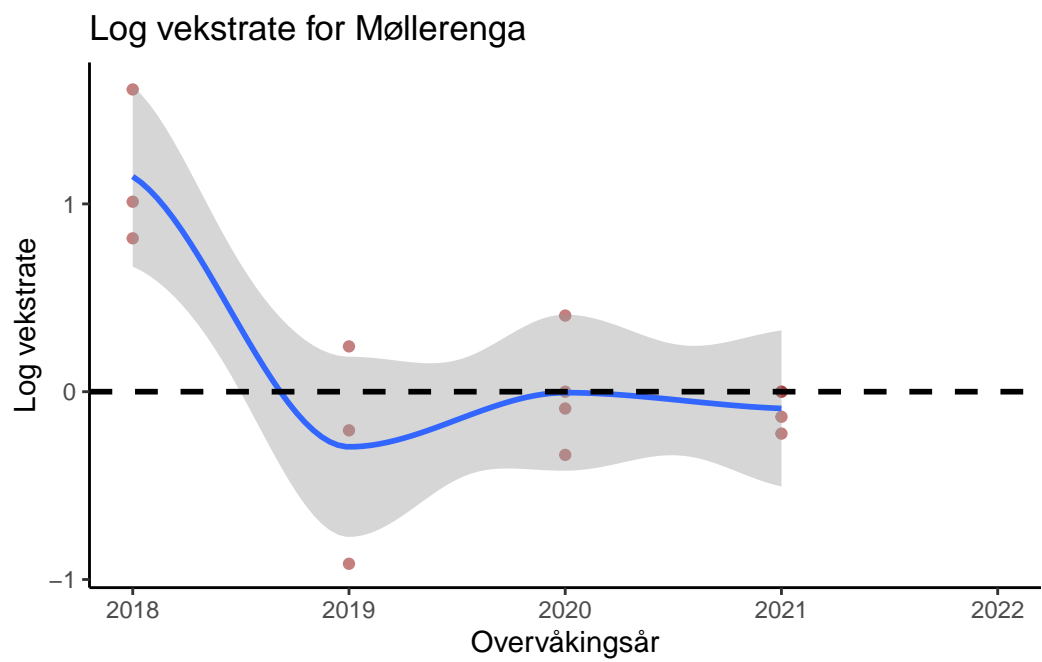
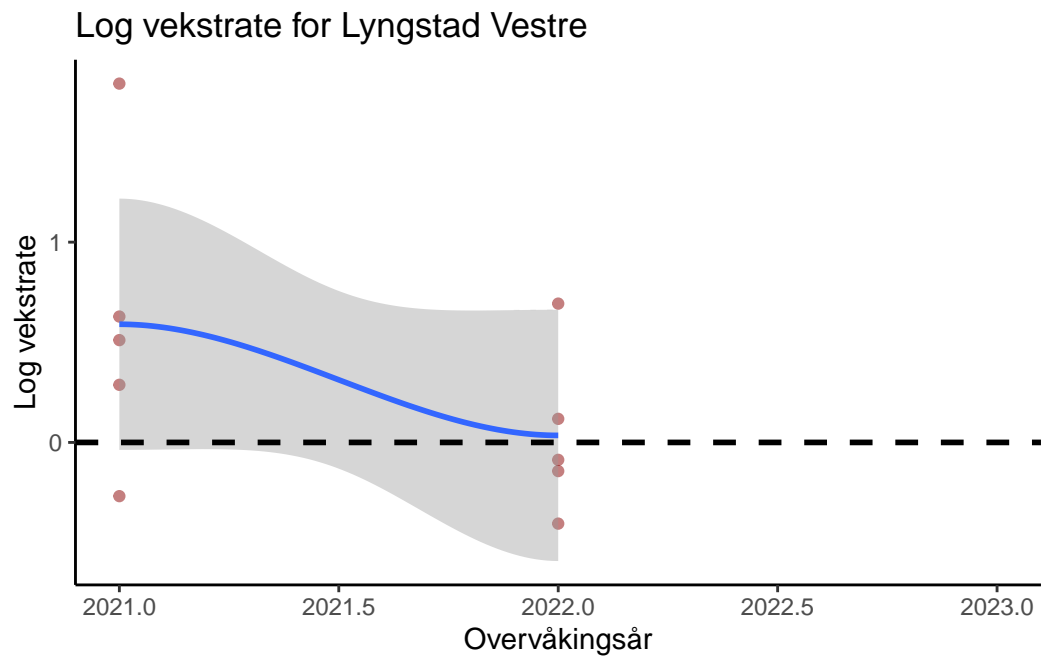


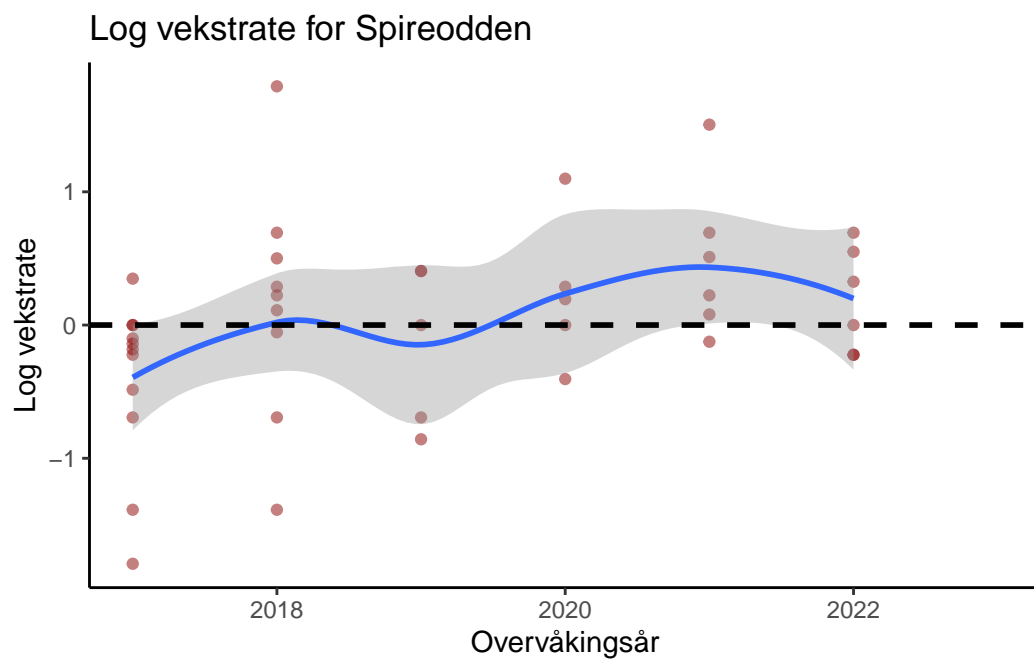
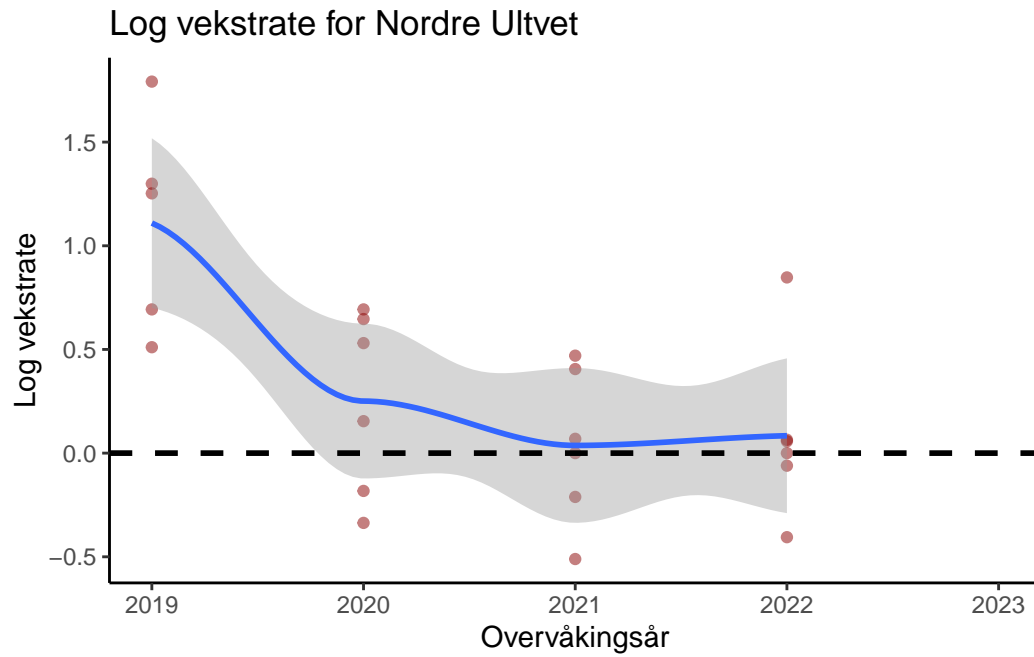


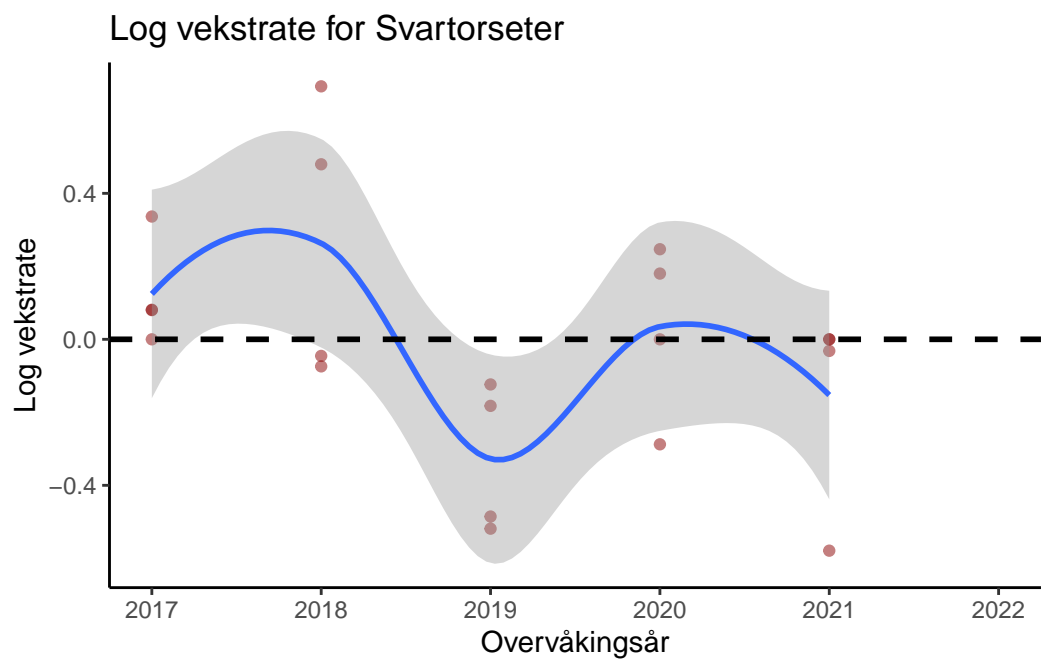
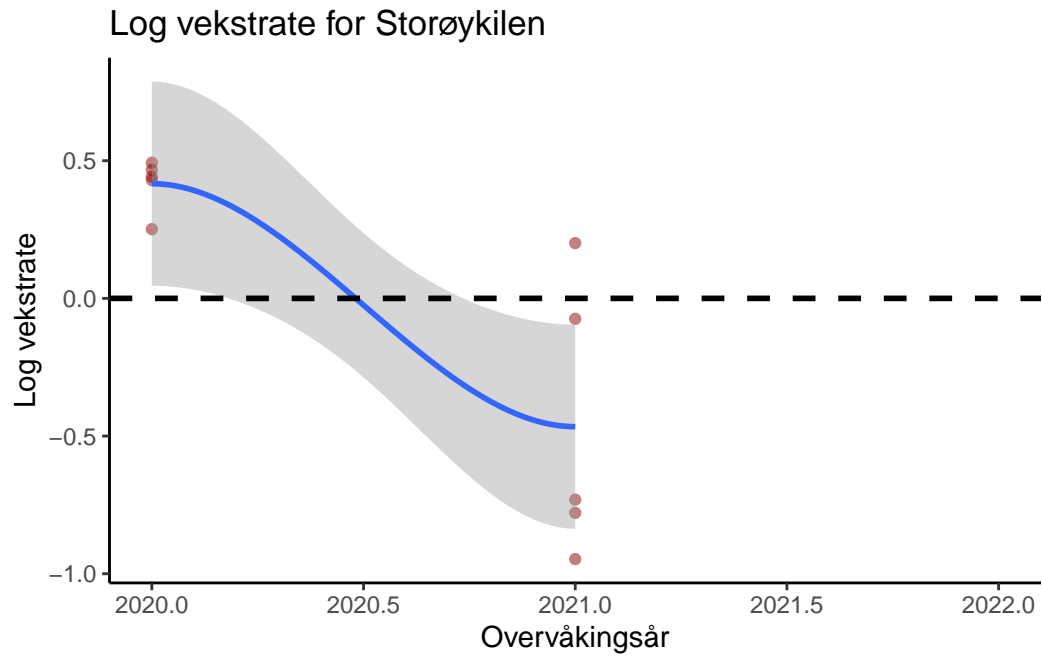


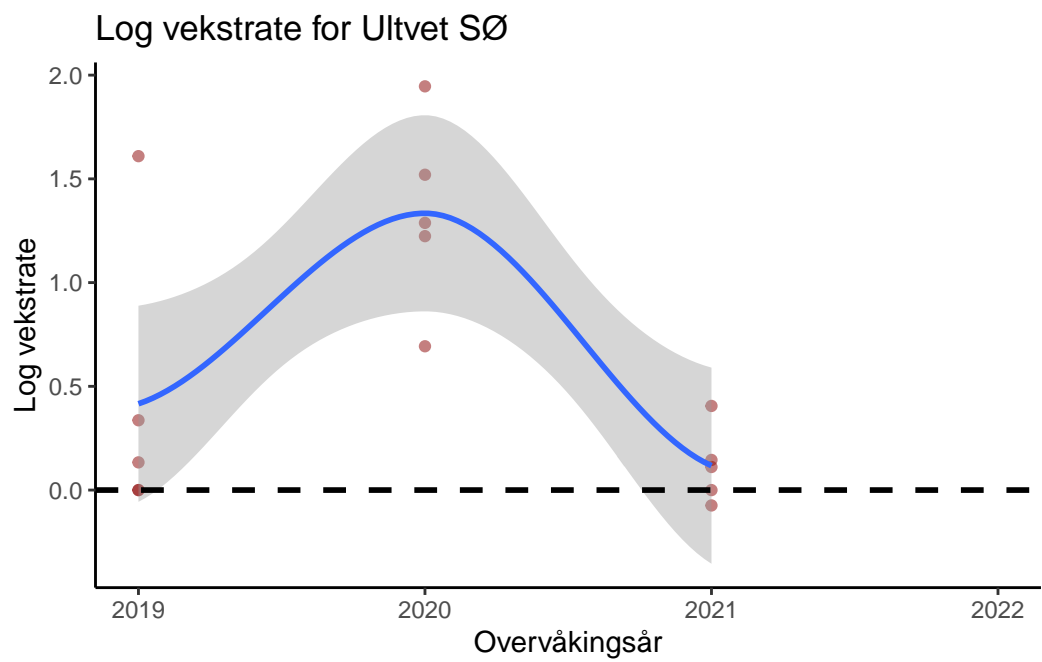
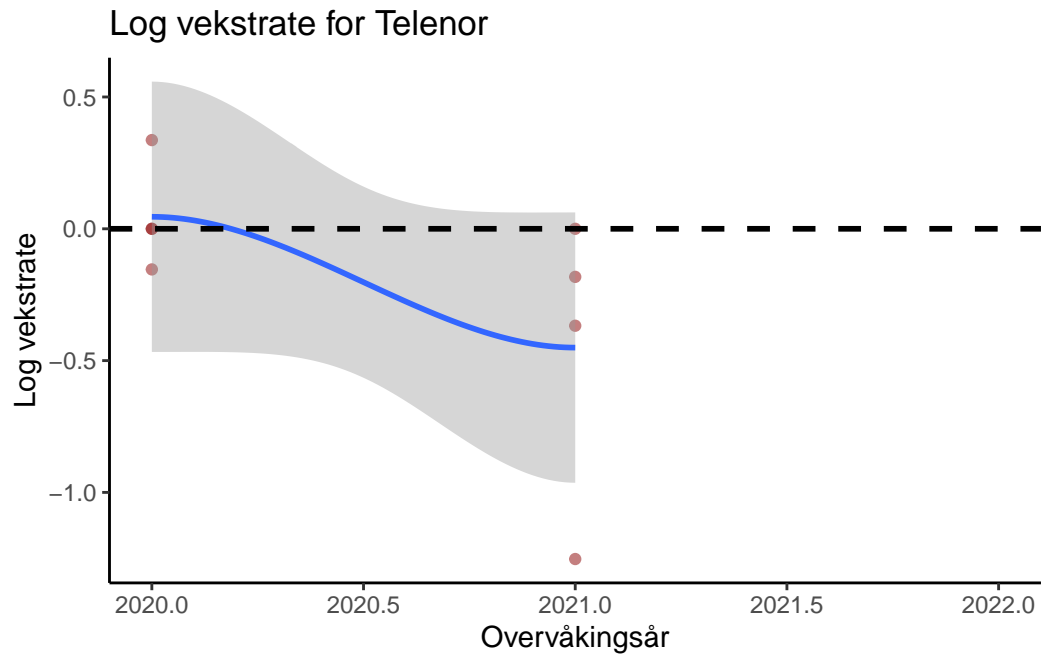


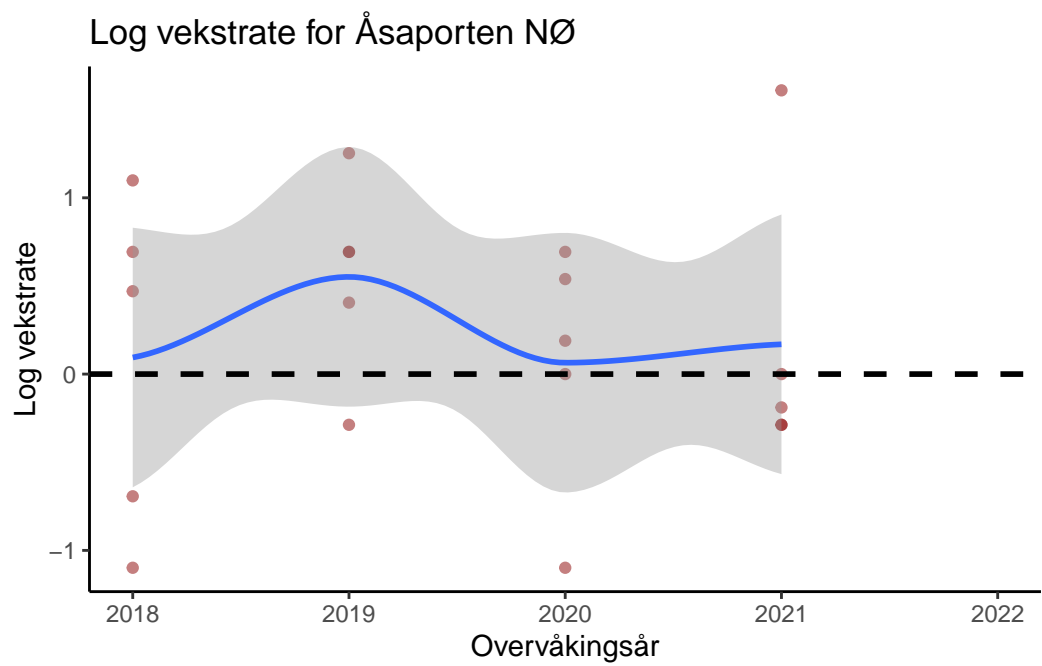
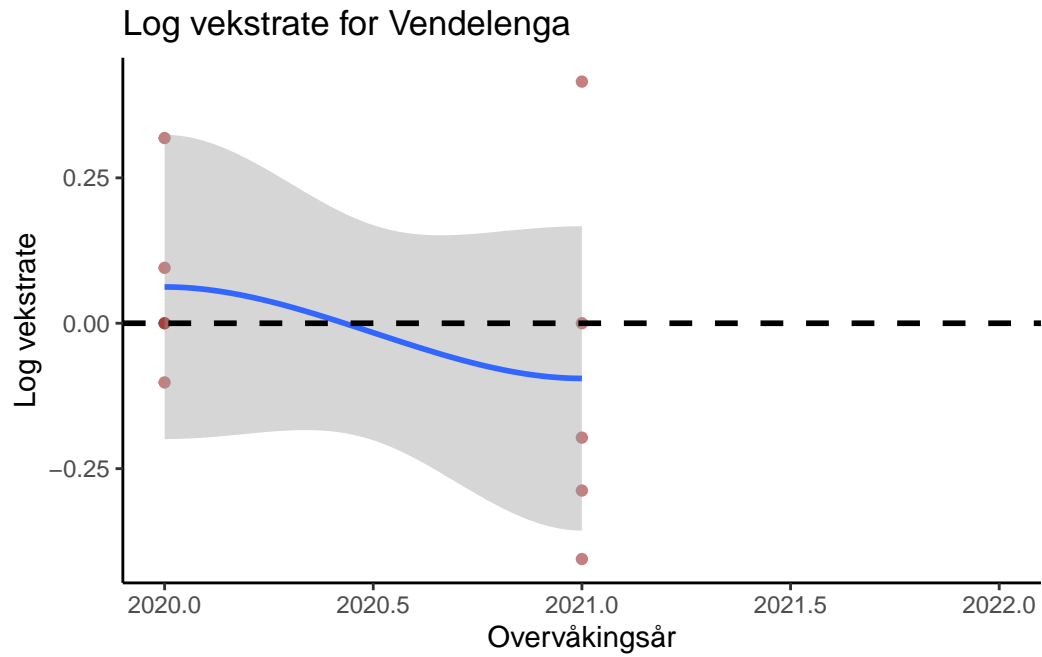










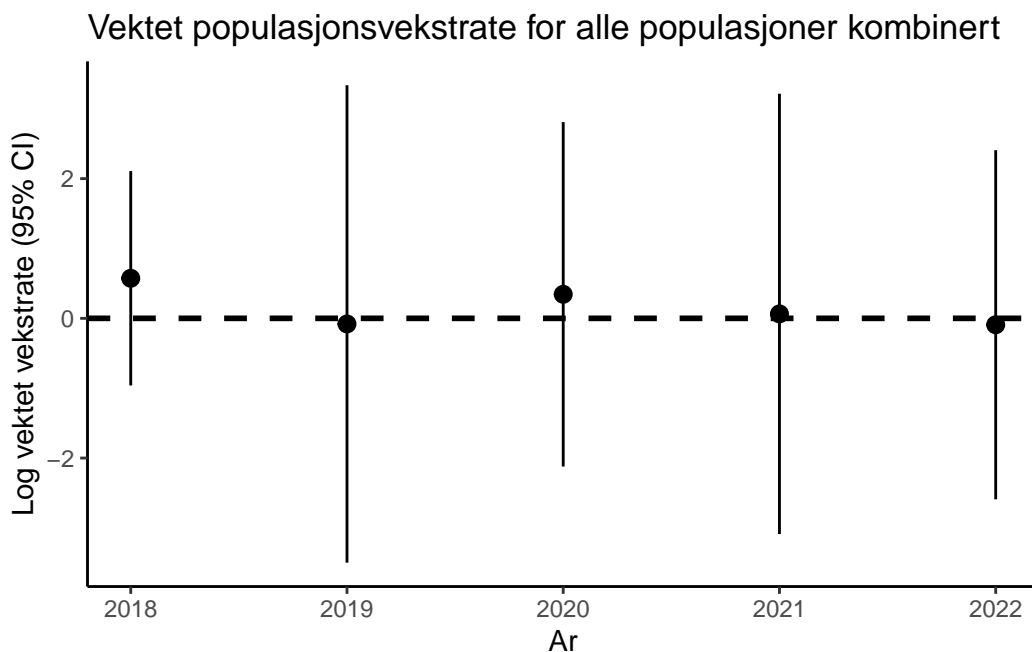


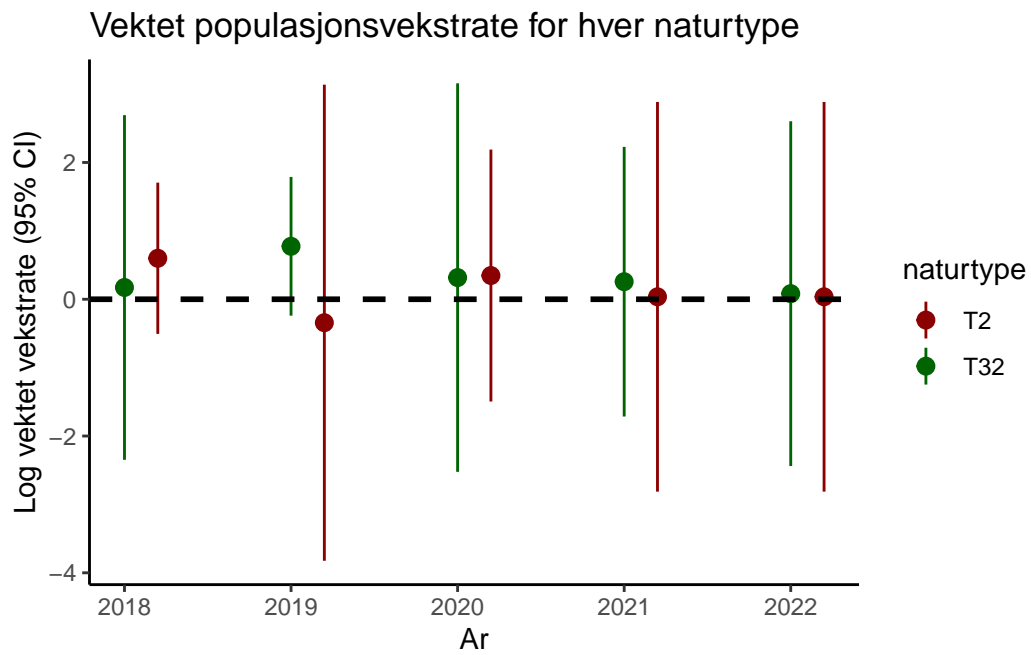
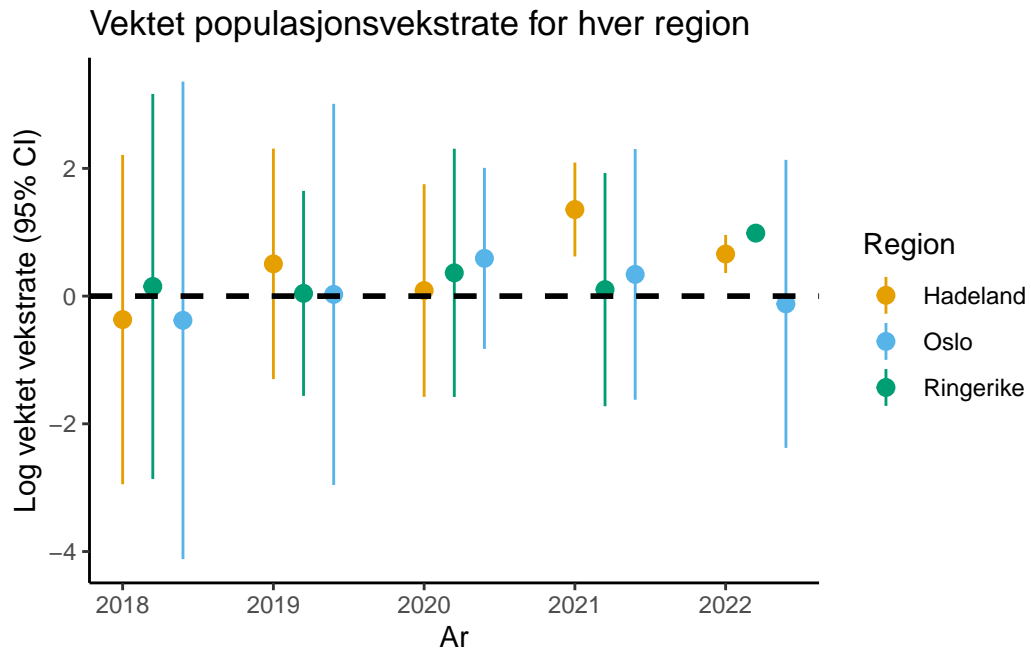
## 4 Vektete populasjonsvekstrater

For å oppsummere vekstratene på tvers av populasjoner brukte vi en vektet regresjon, en tilnærming som brukes i meta-analyser («random effects meta-analysis»). Vi vektet populasjonsvekstratene per lokalitet med den overordnede populasjonsstørrelsen (i den gitte lokaliteten i det gitte året), og et vektet gjennomsnitt ble deretter beregnet totalt (over alle populasjoner), per region og per naturtype.

Denne tilnærmingen gjør at vi kan vekte effekten av endringer i store populasjoner mer, mens endringer i små populasjoner har mindre effekt på overordnede vekstrater. Populasjonsvekstratene gir dermed et representativt estimat på endringer i dragehodepopulasjonen innenfor regionen/naturtypen.

Populasjonsvekstraten er log-transformert. Det betyr at en vekstrate på 0 indikerer en stabil populasjon, vekstrate  $> 0$  indikerer en populasjon i vekst og vekstrate  $< 0$  indikerer en populasjon i nedgang. Figurene viser vektet gjennomsnitt med 95 % konfidensintervall.







## 5 Direction plots

! This is just a dump of plots - needs to be sorted and cleaned up a bit!!!!

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.3      v readr      2.1.4
v forcats    1.0.0      v stringr    1.5.0
v ggplot2    3.4.3      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.0
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(bayestestR)
library(see)
library(rstanarm)
```

Loading required package: Rcpp

This is rstanarm version 2.26.1

- See <https://mc-stan.org/rstanarm/articles/priors> for changes to default priors!
- Default priors may change, so it's safest to specify priors, even if equivalent to the default
- For execution on a local, multicore CPU with excess RAM we recommend calling  
`options(mc.cores = parallel::detectCores())`

```
lokalitetsestimator <- readRDS("C:/Users/matthew.grainger/Documents/Projects_in_development/Dra
lokalitetsdata <- readRDS("C:/Users/matthew.grainger/Documents/Projects_in_development/Dra
```

```

#length(lokalitetsestimator)
#seq_along(lokalitetsestimator)

for (i in seq_along(lokalitetsestimator)){
  dlist=pluck(lokalitetsestimator,i)
  df=data.frame(year=dlist$year, nTot=dlist$nTot, nTotUpp=dlist$Tot.CI[,2],
                nTotLow=dlist$Tot.CI[,1], nFert=dlist$nFert,
                nFertUpp=dlist$Fert.CI[,2],
                nFertLow=dlist$Fert.CI[,1],
                nVeg=dlist$nVeg,
                nVegUpp=dlist$Veg.CI[,2],
                nVegLow=dlist$Veg.CI[,1],
                nSma=dlist$nSma,
                nSmaUpp=dlist$Sma.CI[,2],
                nSmaLow=dlist$Sma.CI[,1],
                lokalitet=dlist$lokalitet)

  saveRDS(df,
           paste0("data/derived_data/direction_plots/",dlist$lokalitet, "_data.RDS"))
}

rds_combo <- list.files(path = "data/derived_data/direction_plots" , pattern = "*.RDS", full
  map_dfr(readRDS)

#names(rds_combo)

#rds_combo

local=lokalitetsdata %>%
  select(Region, Kommune, Lokalitet) %>%
  rename(lokalitet=Lokalitet)
rds_combo=rds_combo %>%
  left_join(
    .,local
  )

for (i in seq_along(lokalitetsestimator)){
  tryCatch(
    {
      lokalitetsnavn=unique(rds_combo$lokalitet)
      lokalitetsnavn_=lokalitetsnavn[i]
      modeldata=rds_combo %>% filter(lokalitet==lokalitetsnavn_) |> distinct()
    }
  )
}

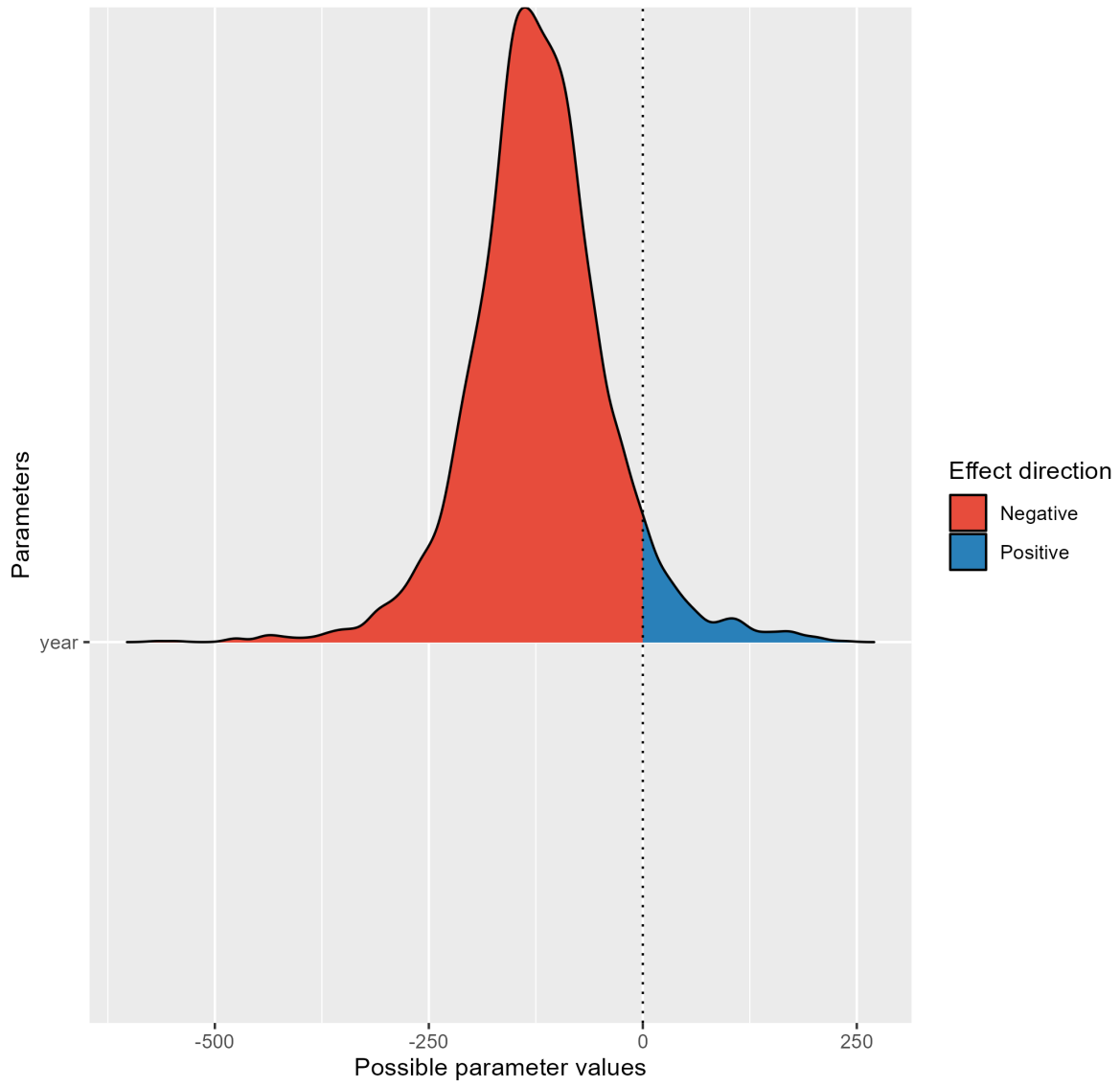
```

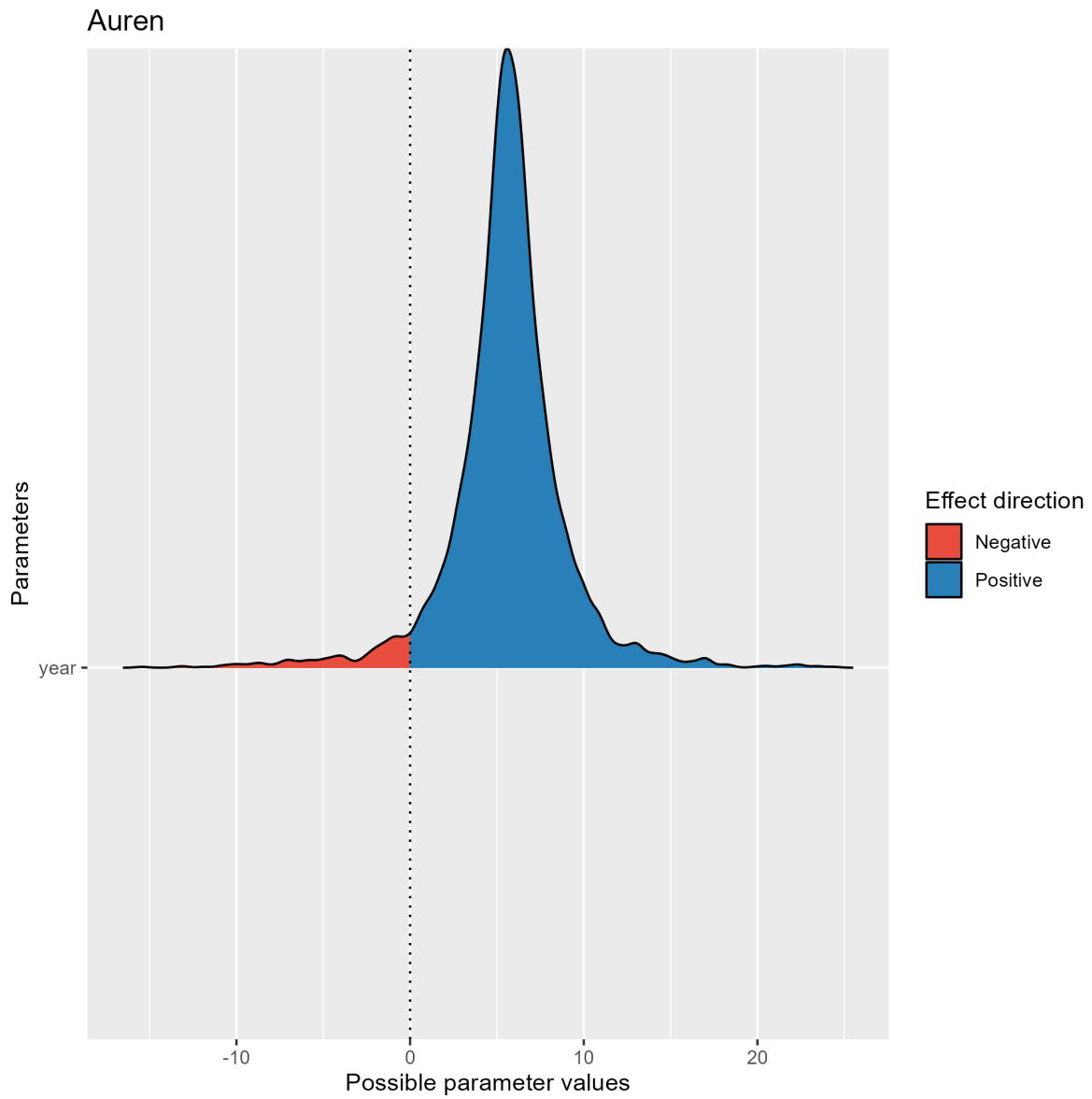
```

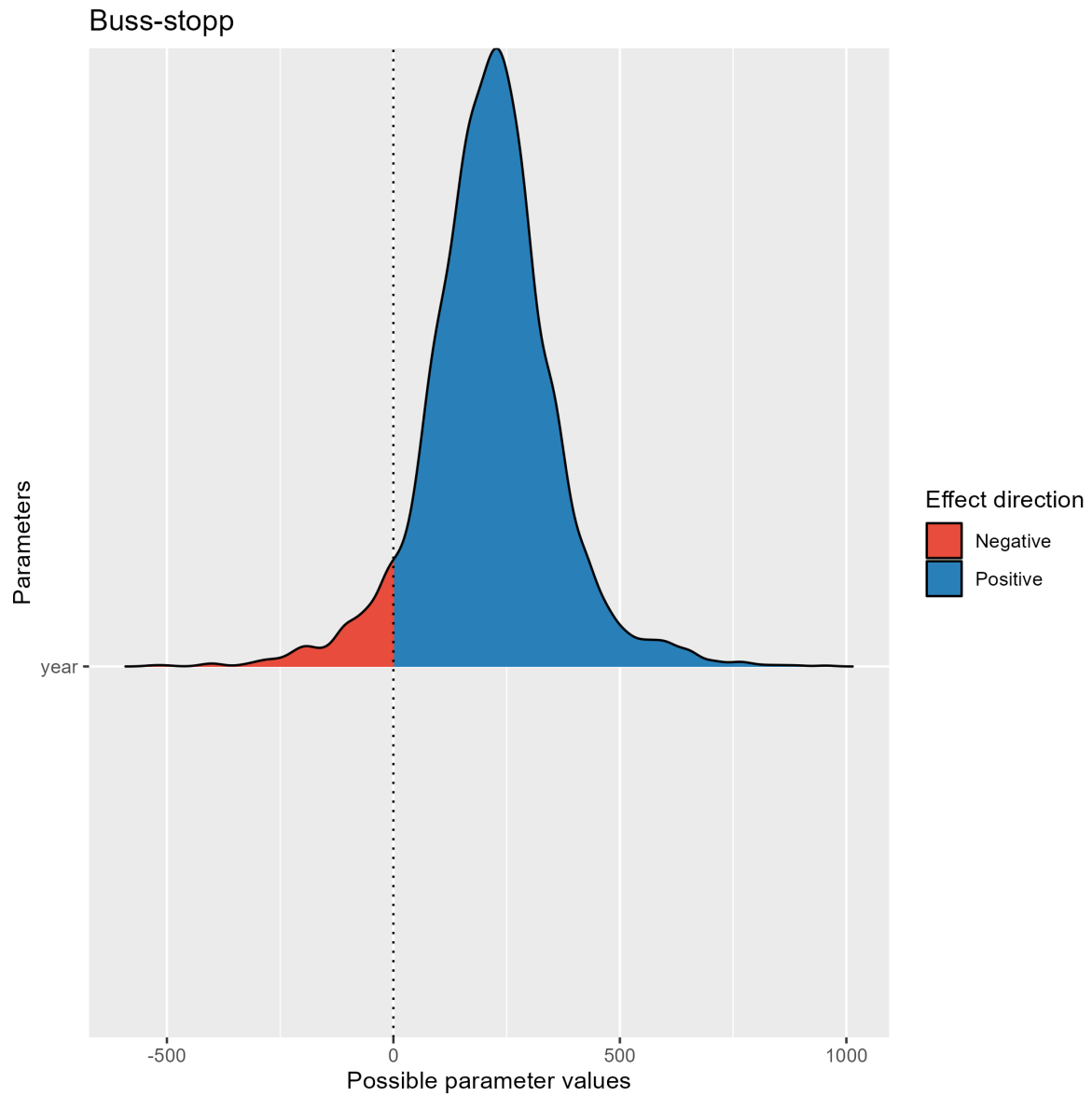
model1<-stan_glm(nTot~year, data=modeldata)
saveRDS(model1, file=paste0("data/derived_data/direction_plots/models/",lokalitetsna
pd=bayestestR::p_direction(model1)
p=plot(pd)+ggtitle(paste0(lokalitetsnavn_))
ggsave(filename=paste0("data/derived_data/direction_plots/plots/", lokalitetsnavn_",
},
error=function(cond) {
  message("Here's the original error message:")
  message(cond)
  # Choose a return value in case of error
  return(NA)
},
warning=function(cond) {
  message("Here's the original warning message:")
  message(cond)
  # Choose a return value in case of warning
  return(NULL)
}
})

```

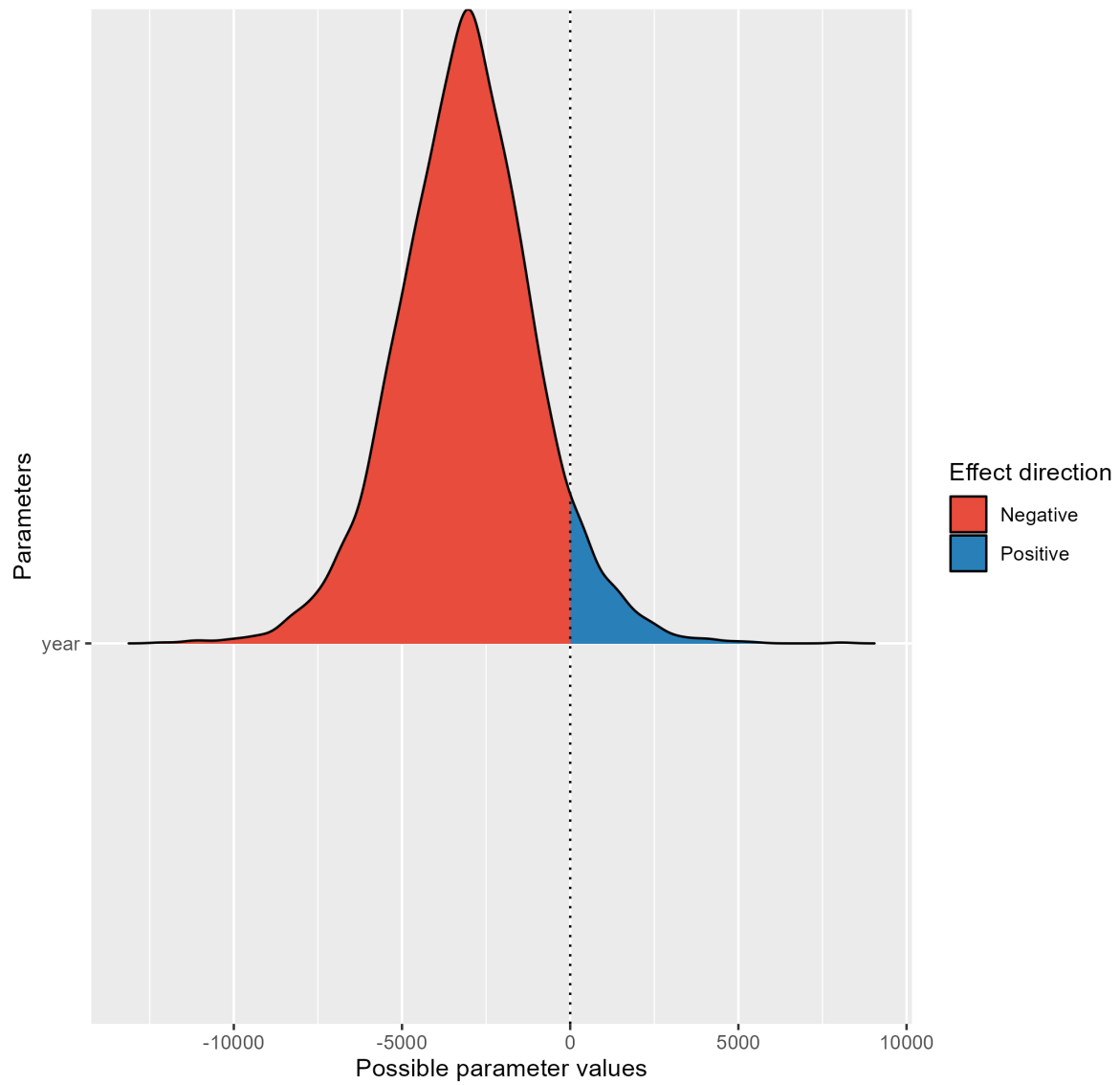
Aslaksrud

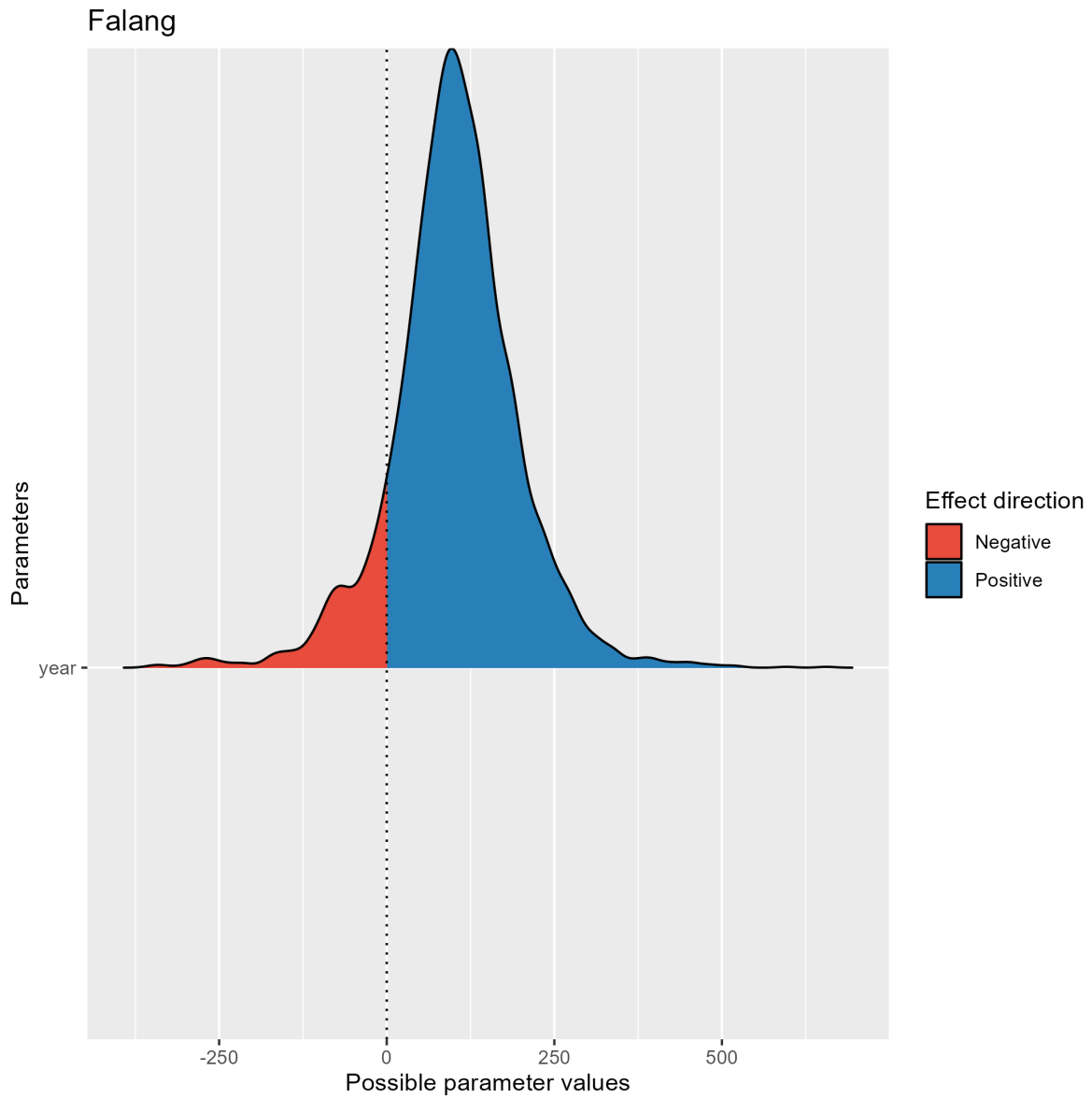




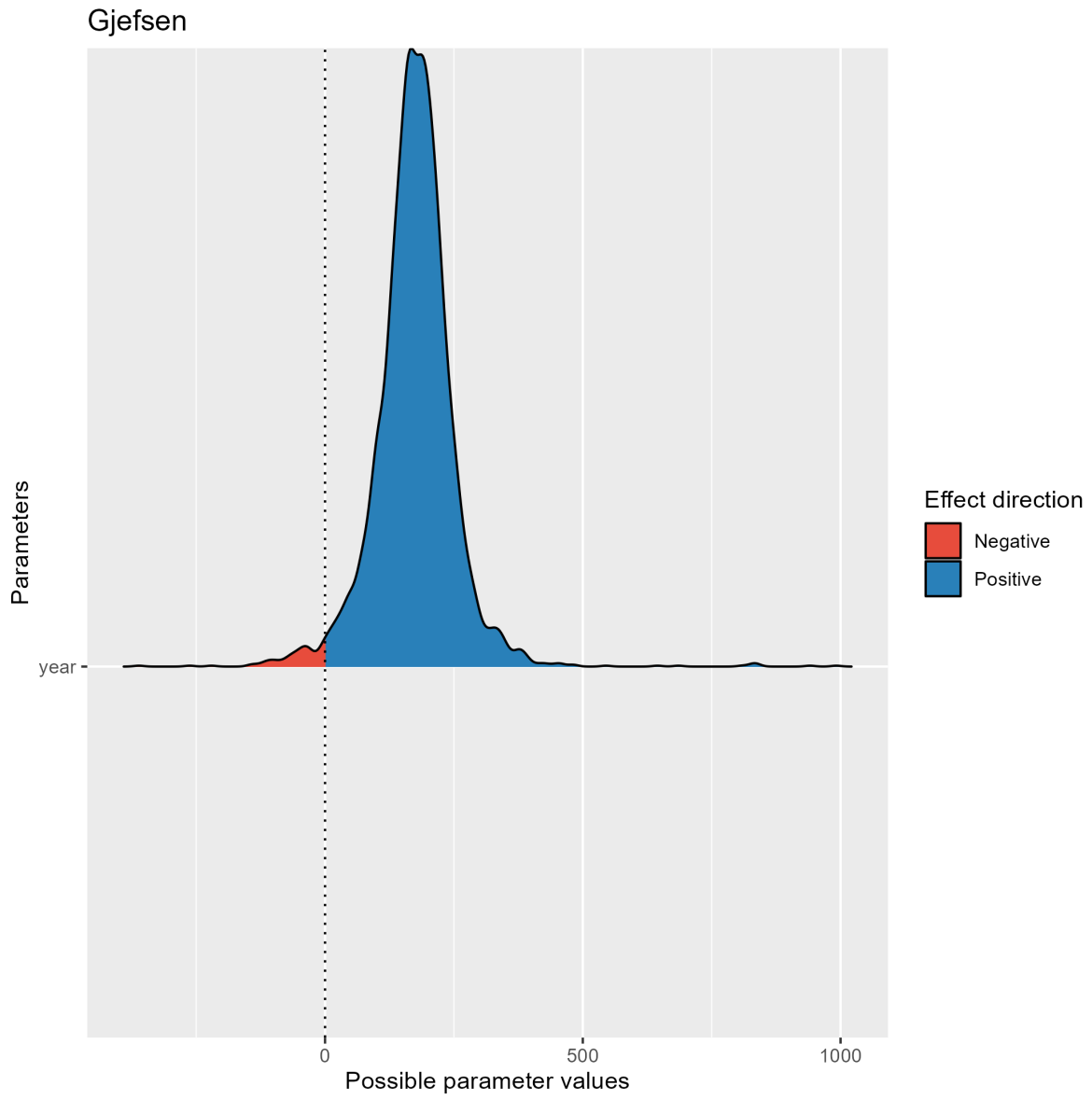


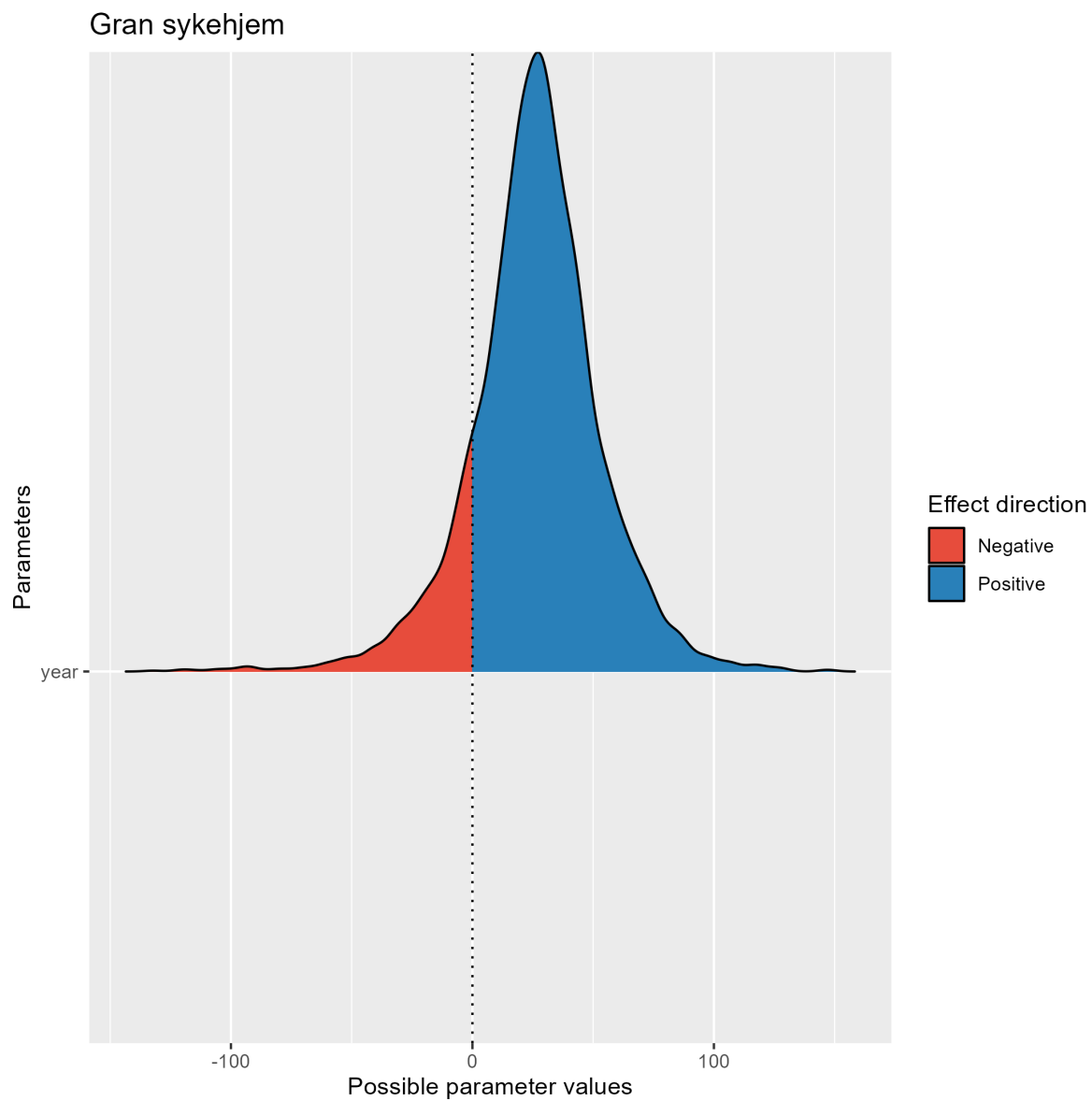
# Ekebergskråningen

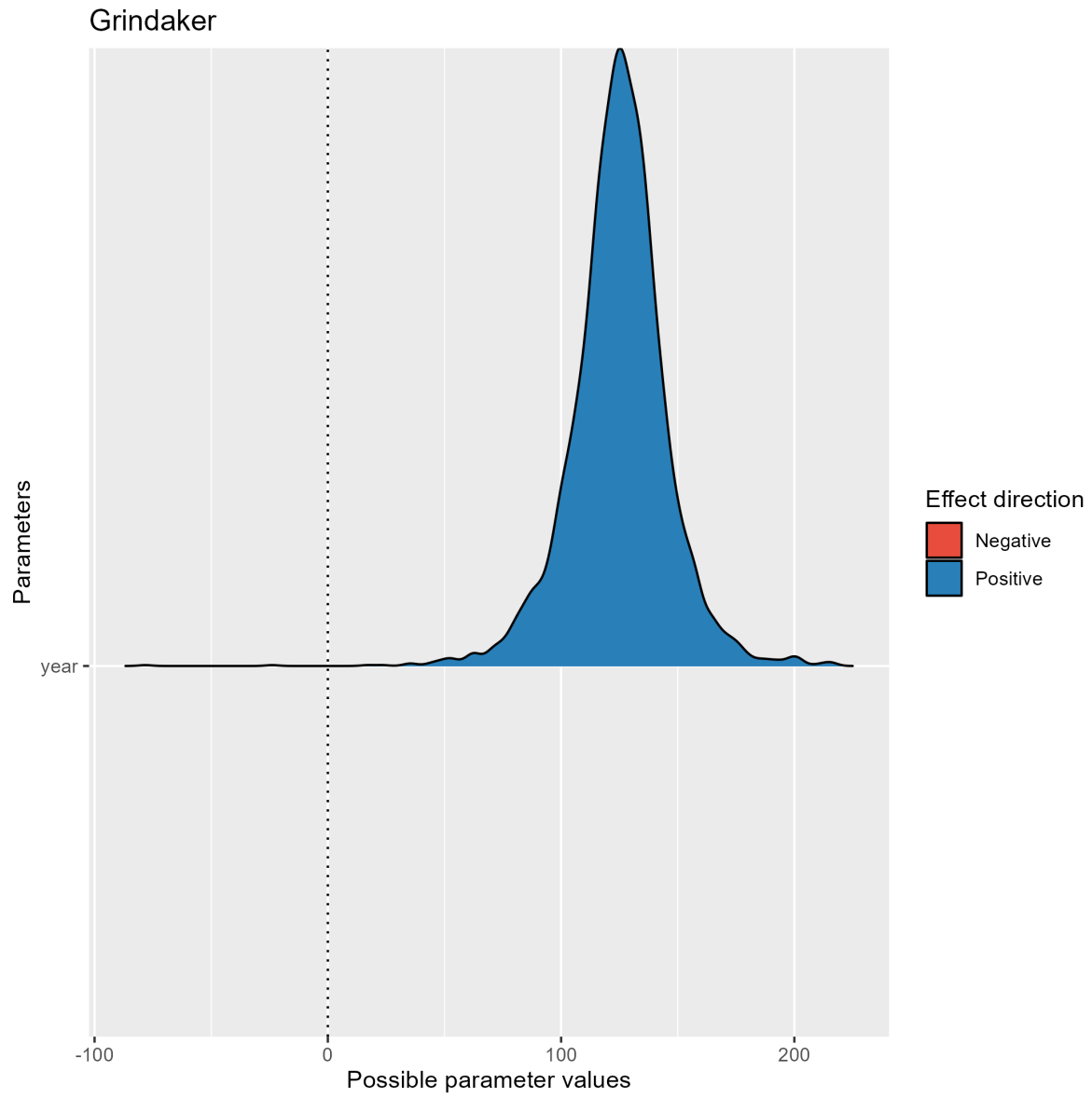




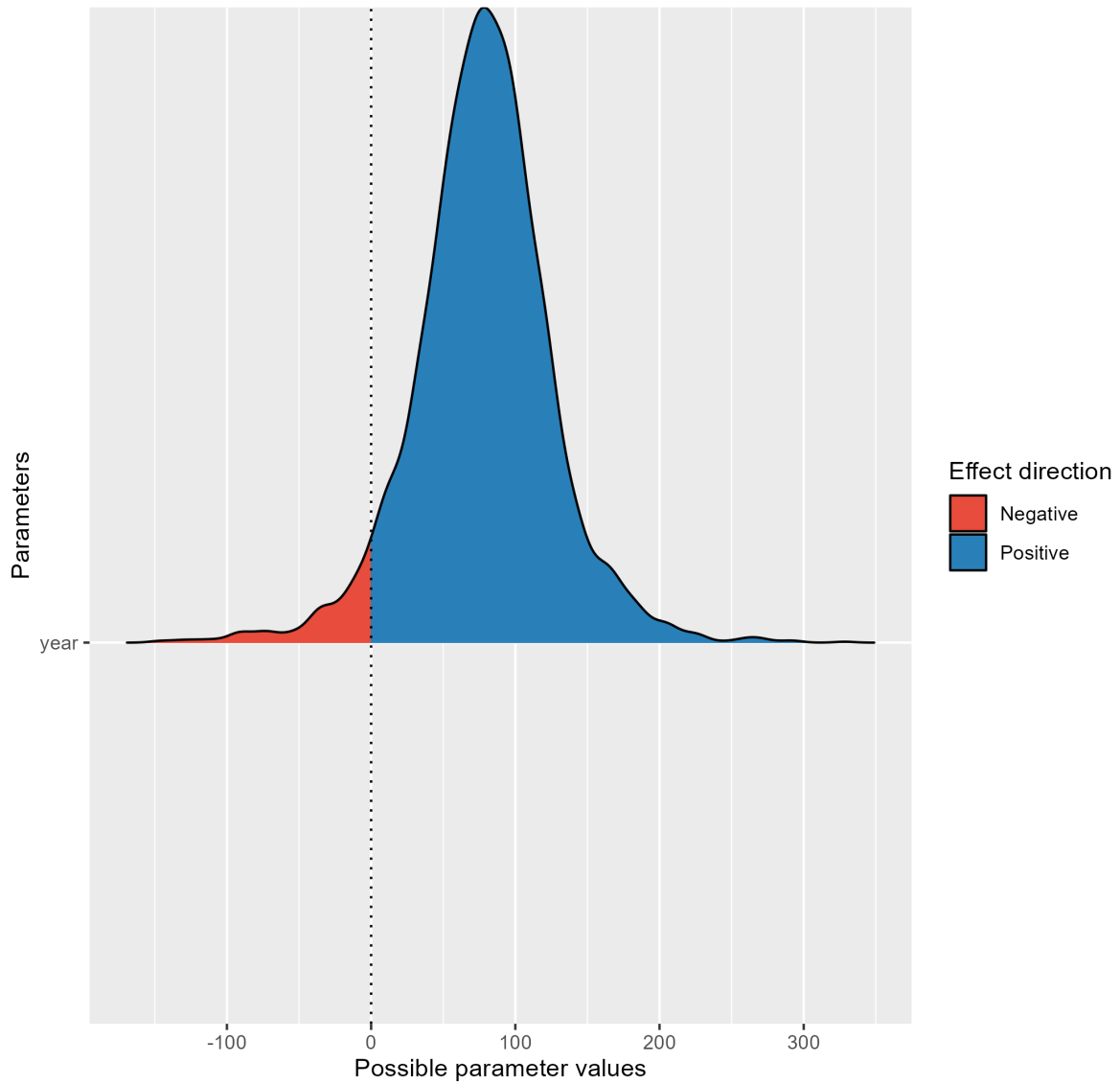


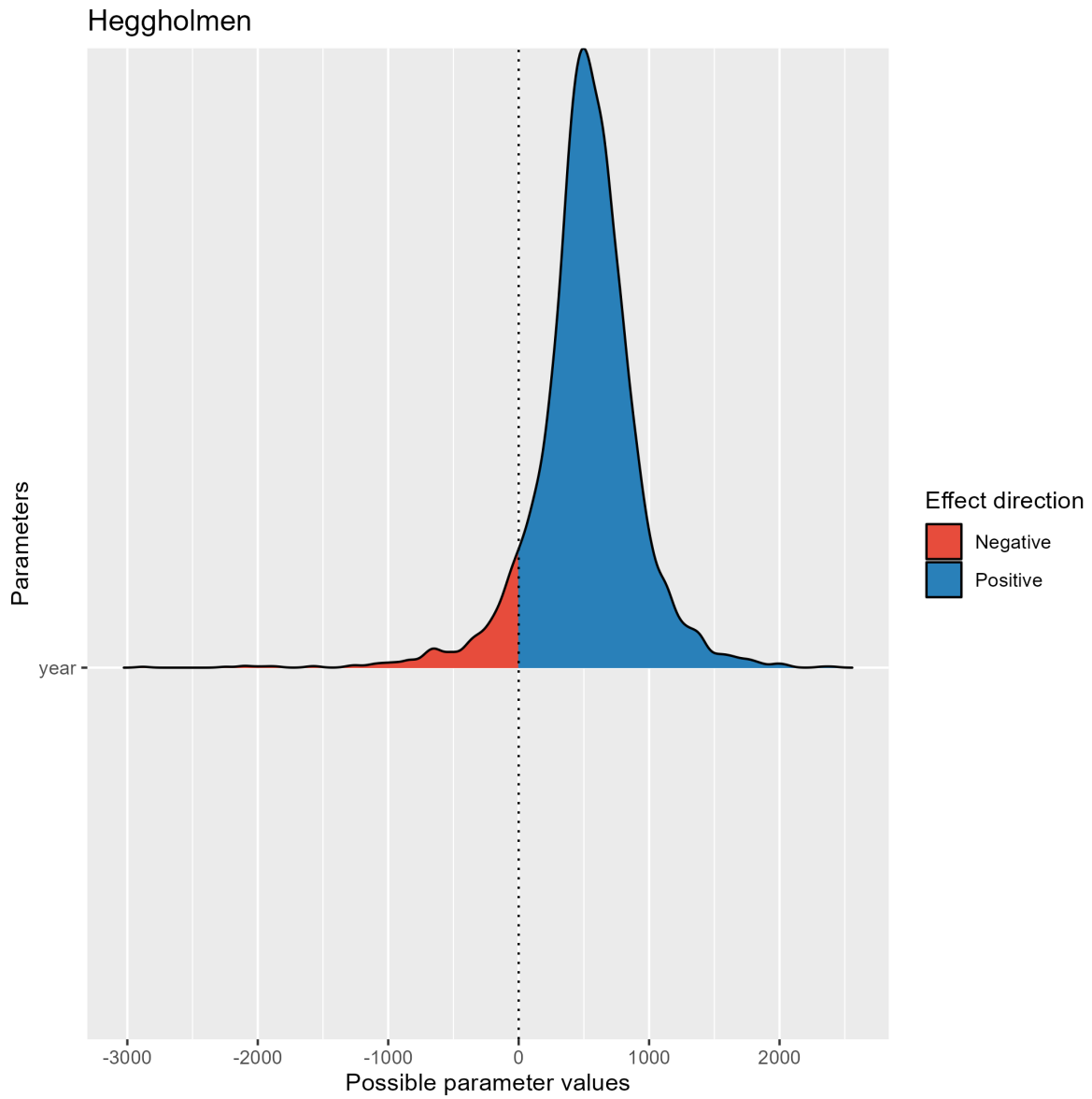


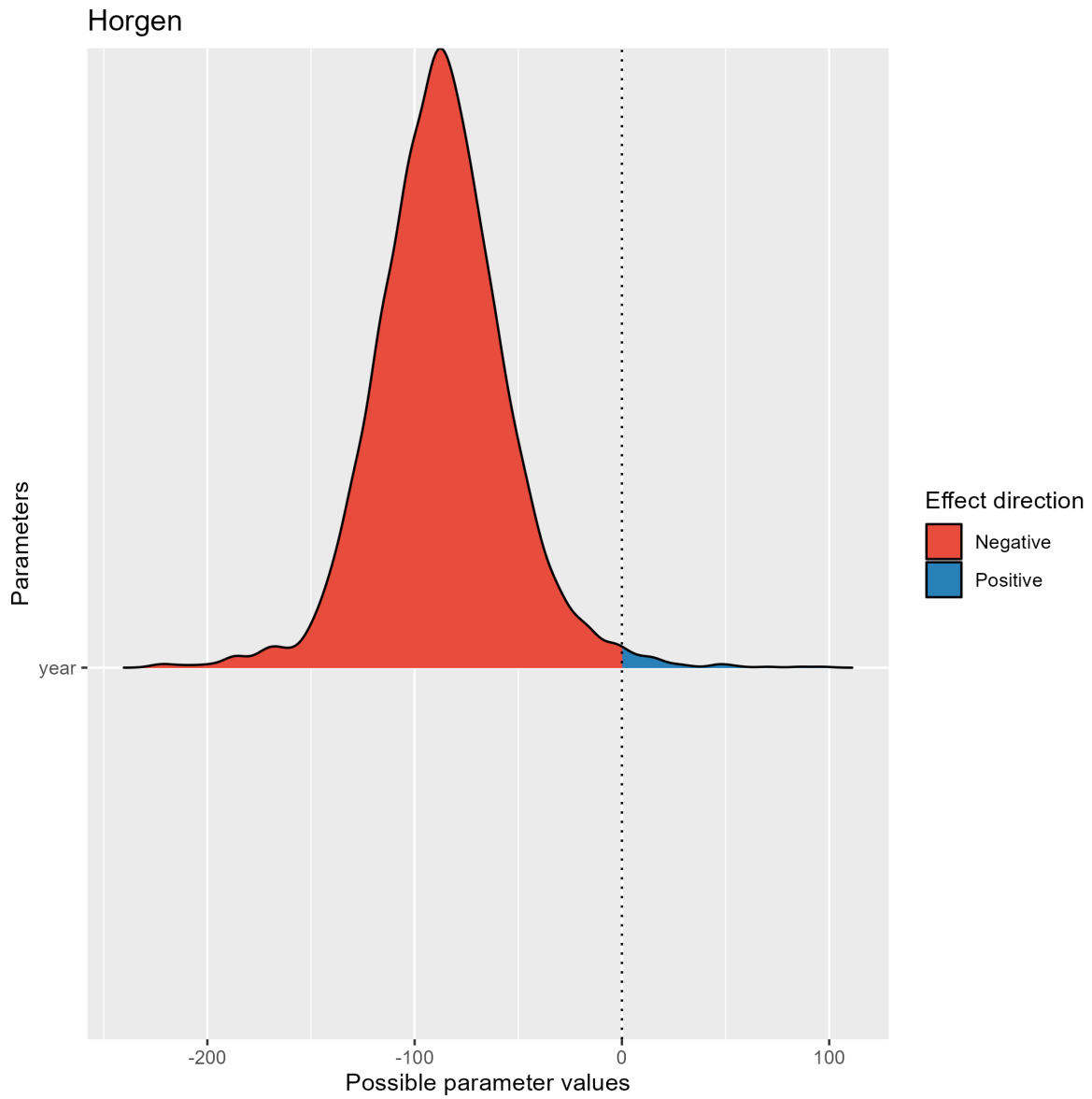




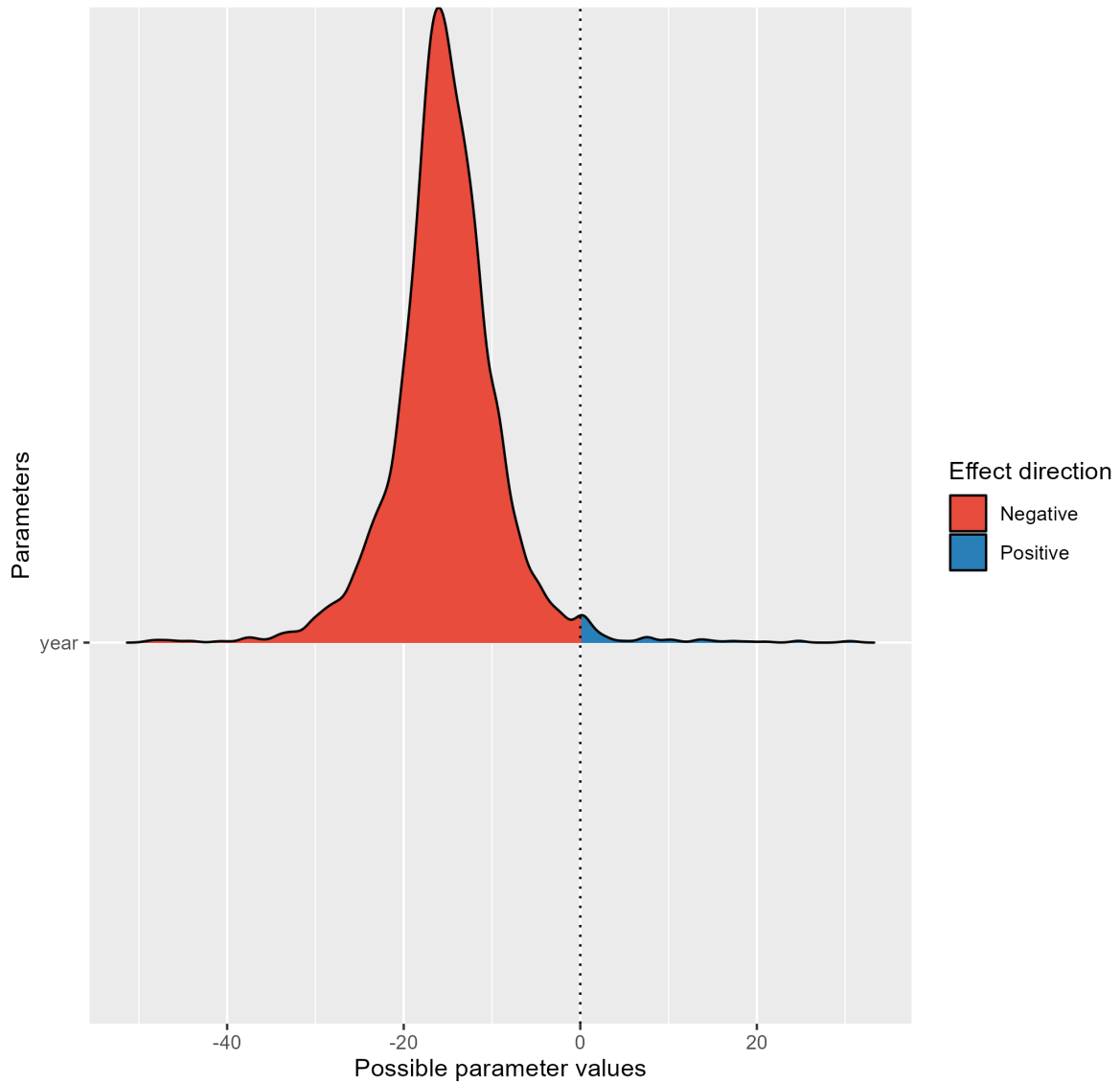
# Haugsbygd, vestvendt vegskråning

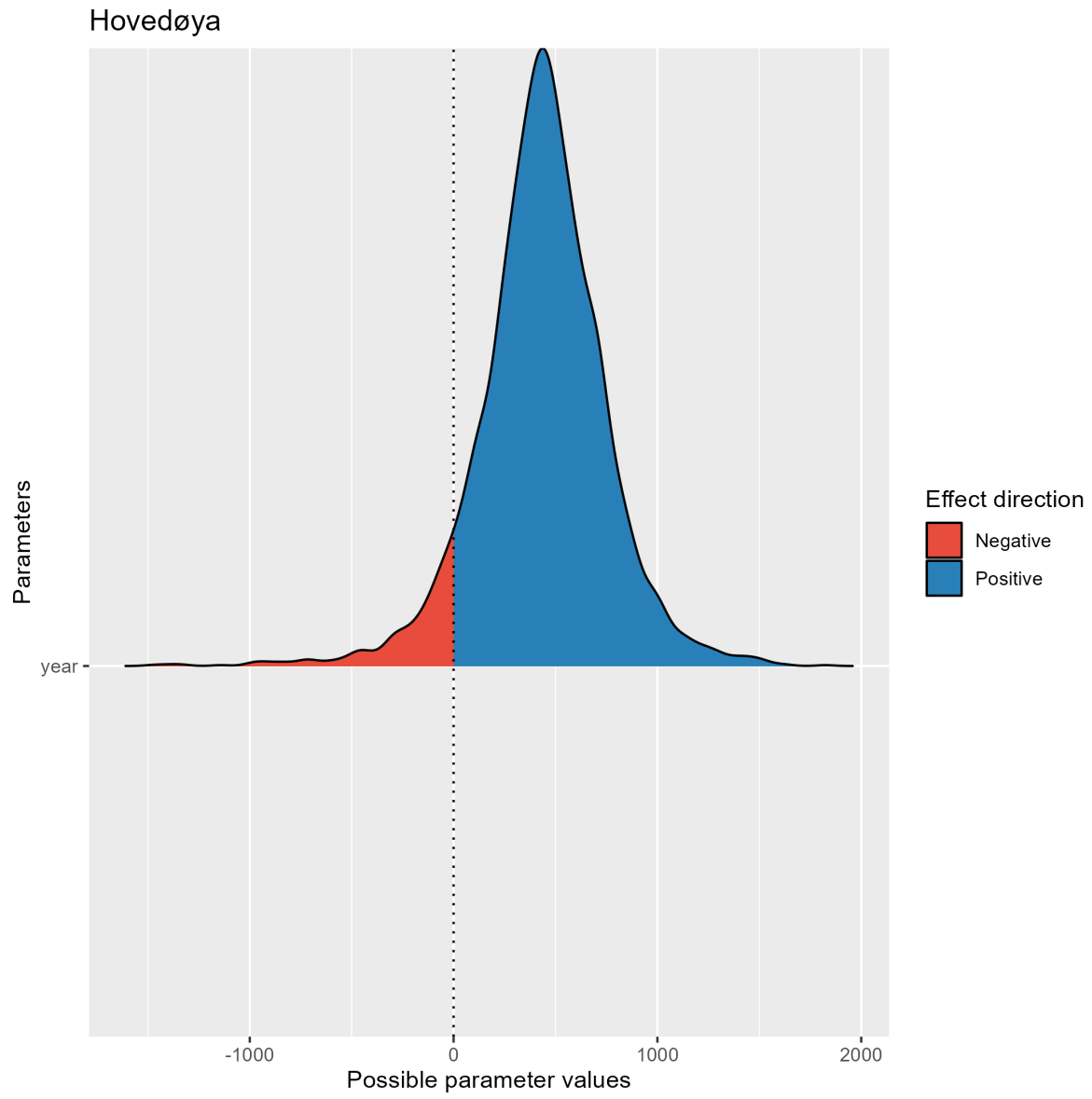




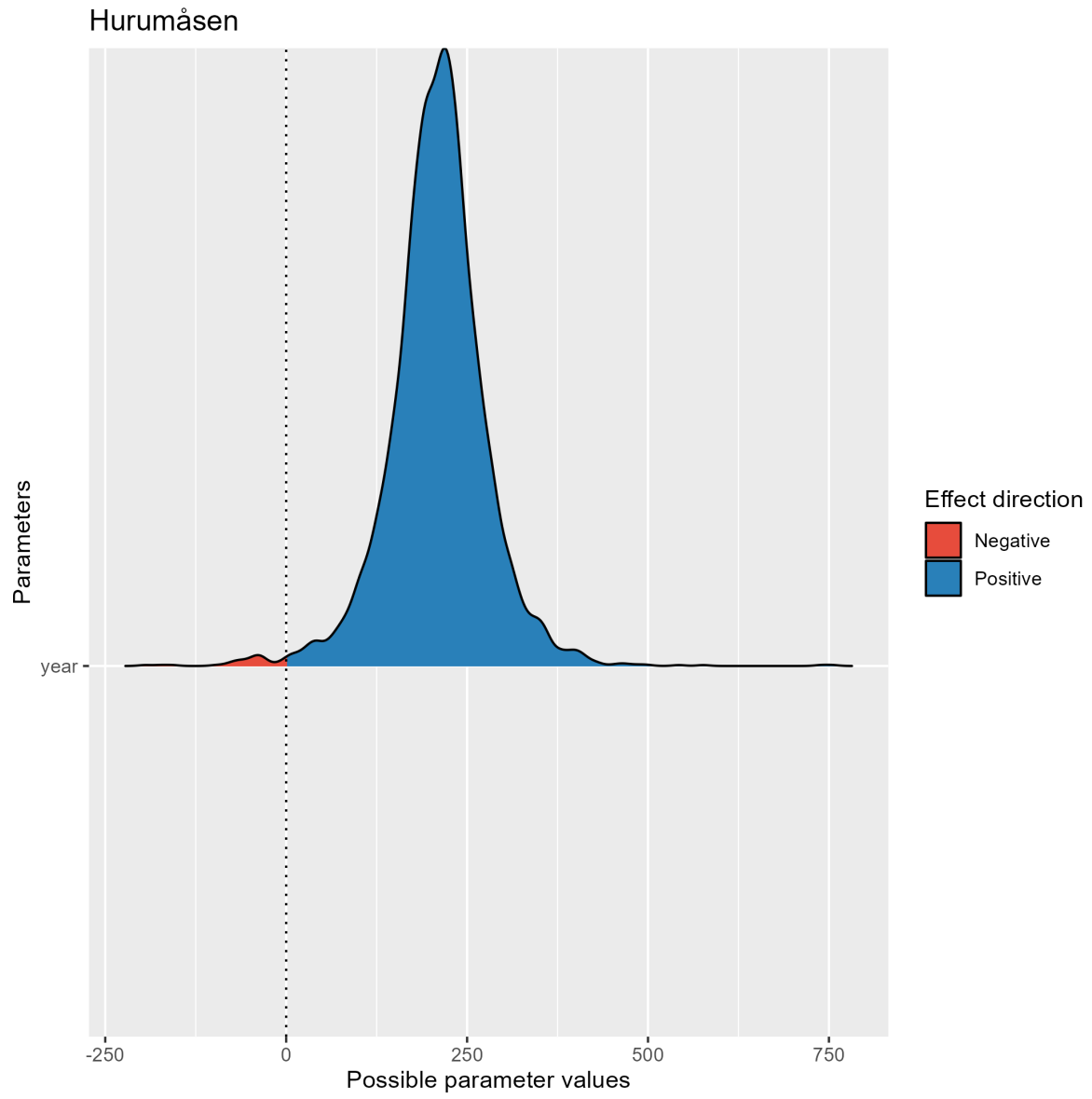


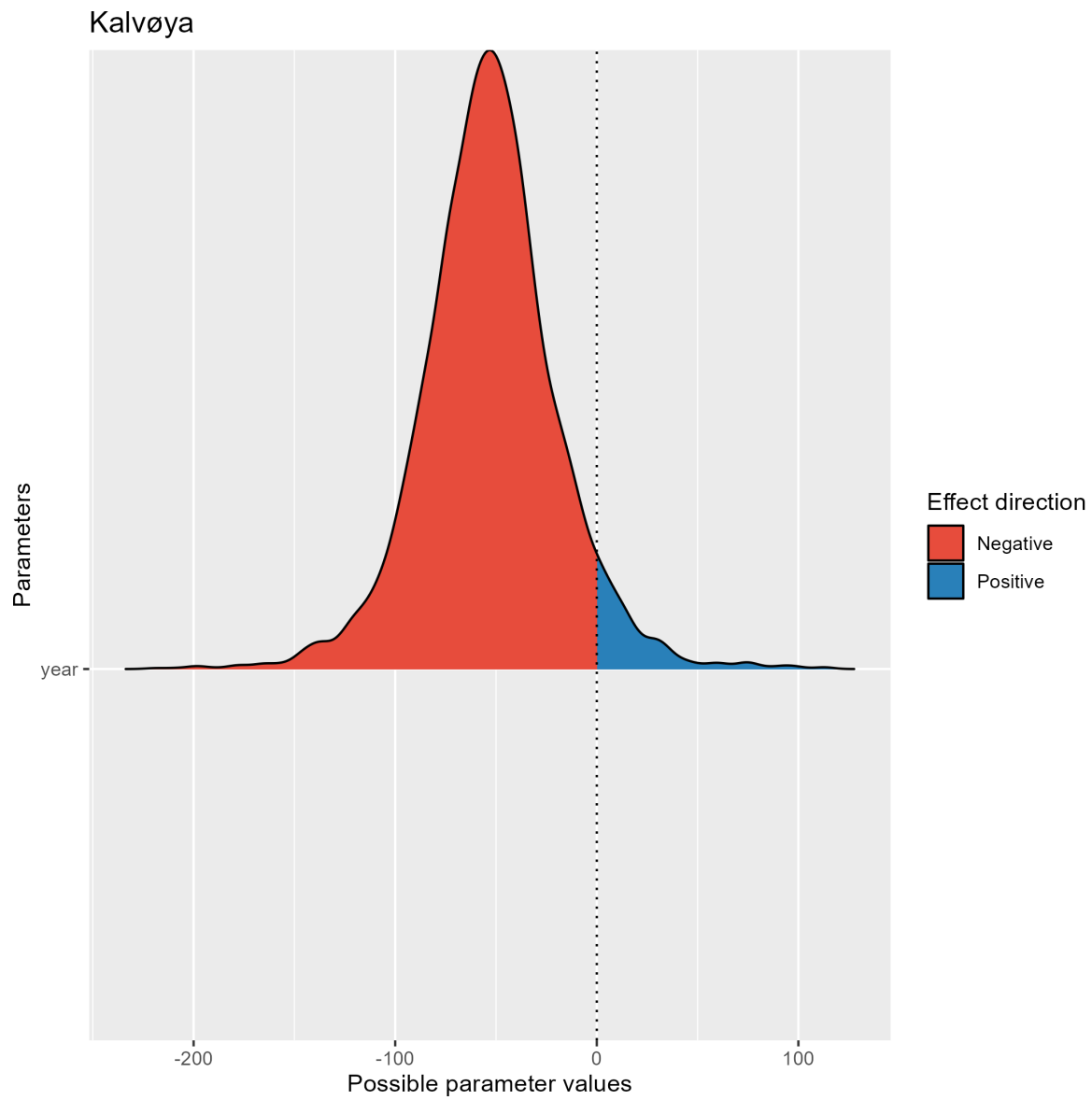
# Hovedøya V

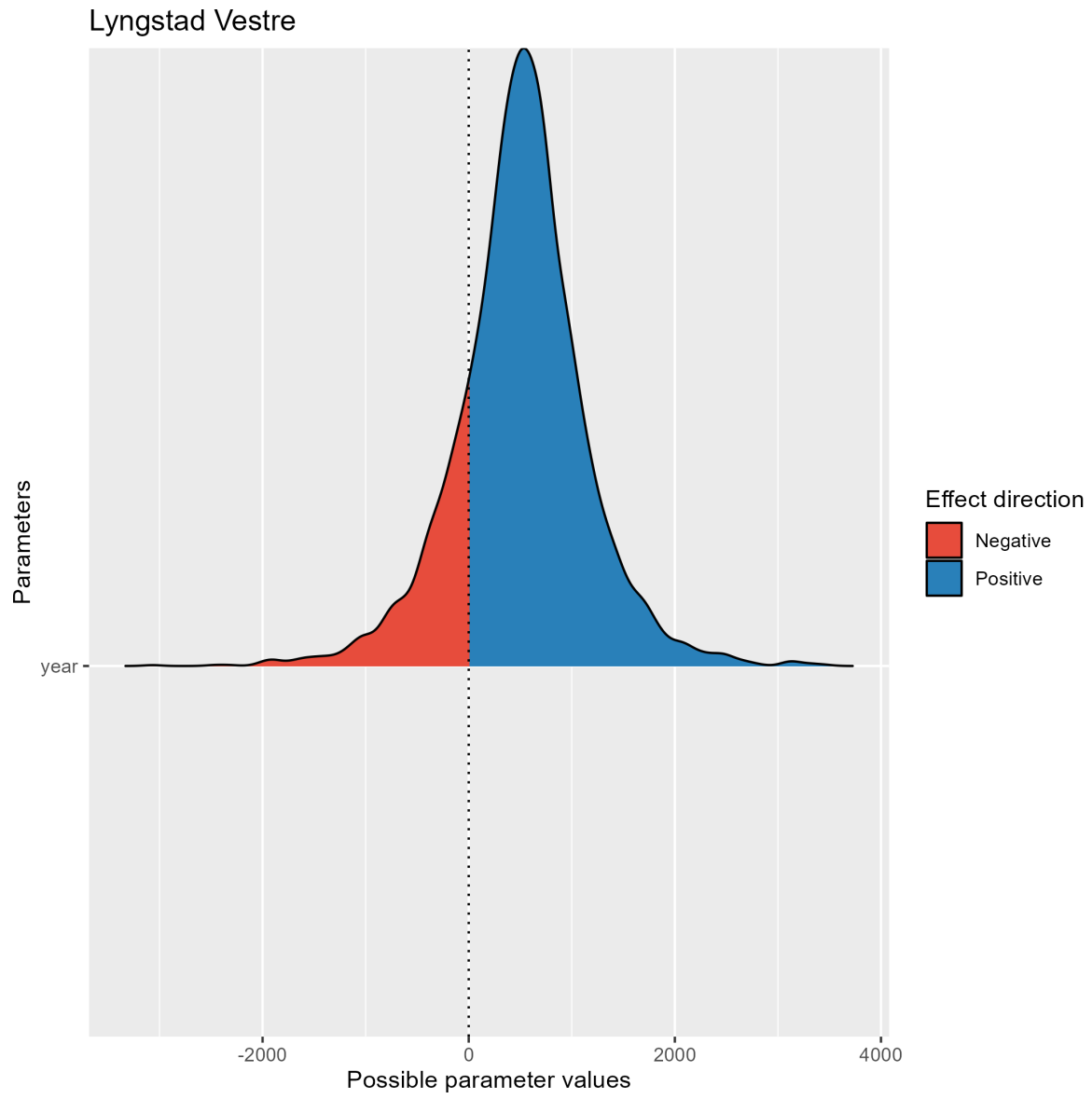


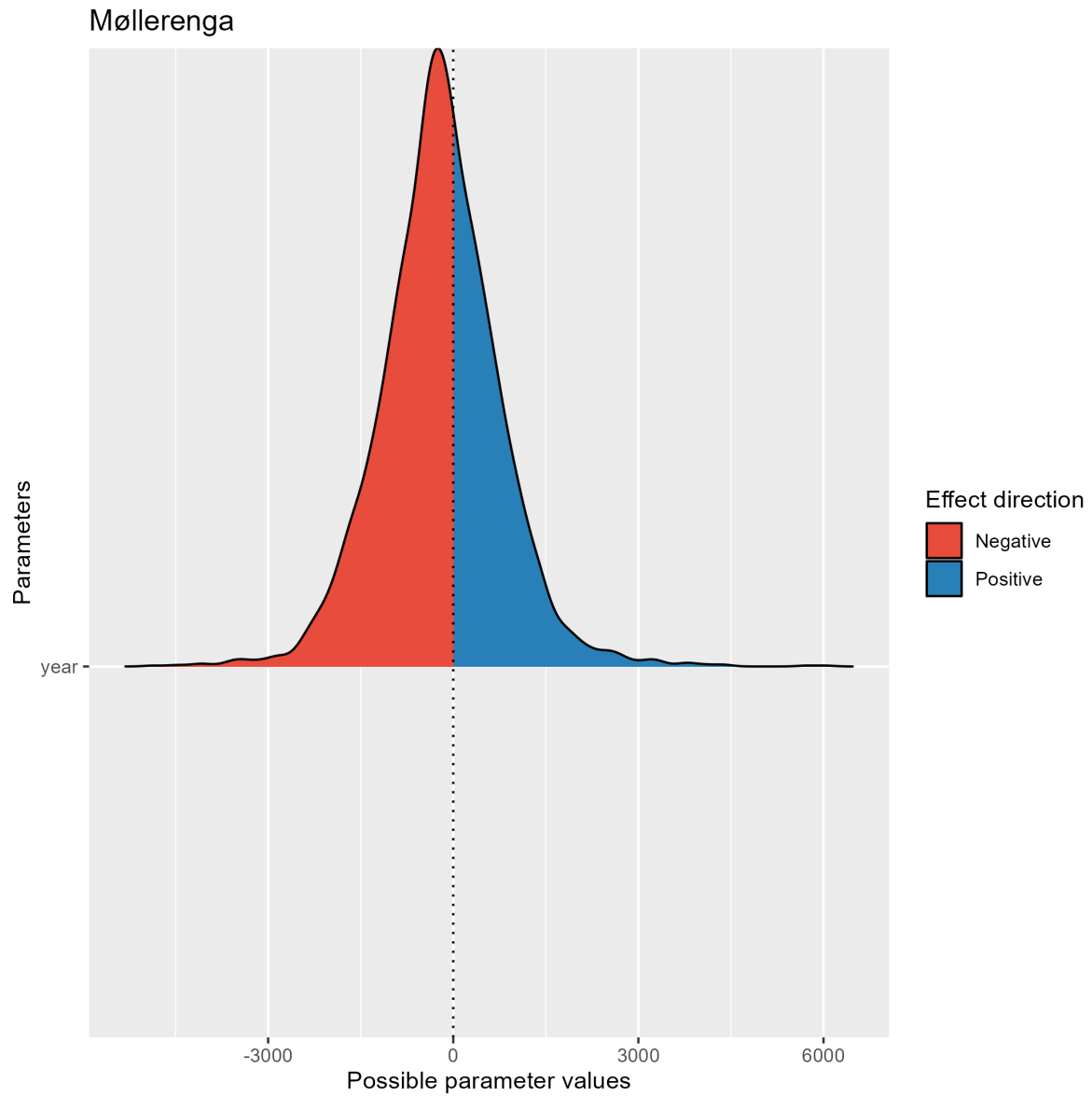


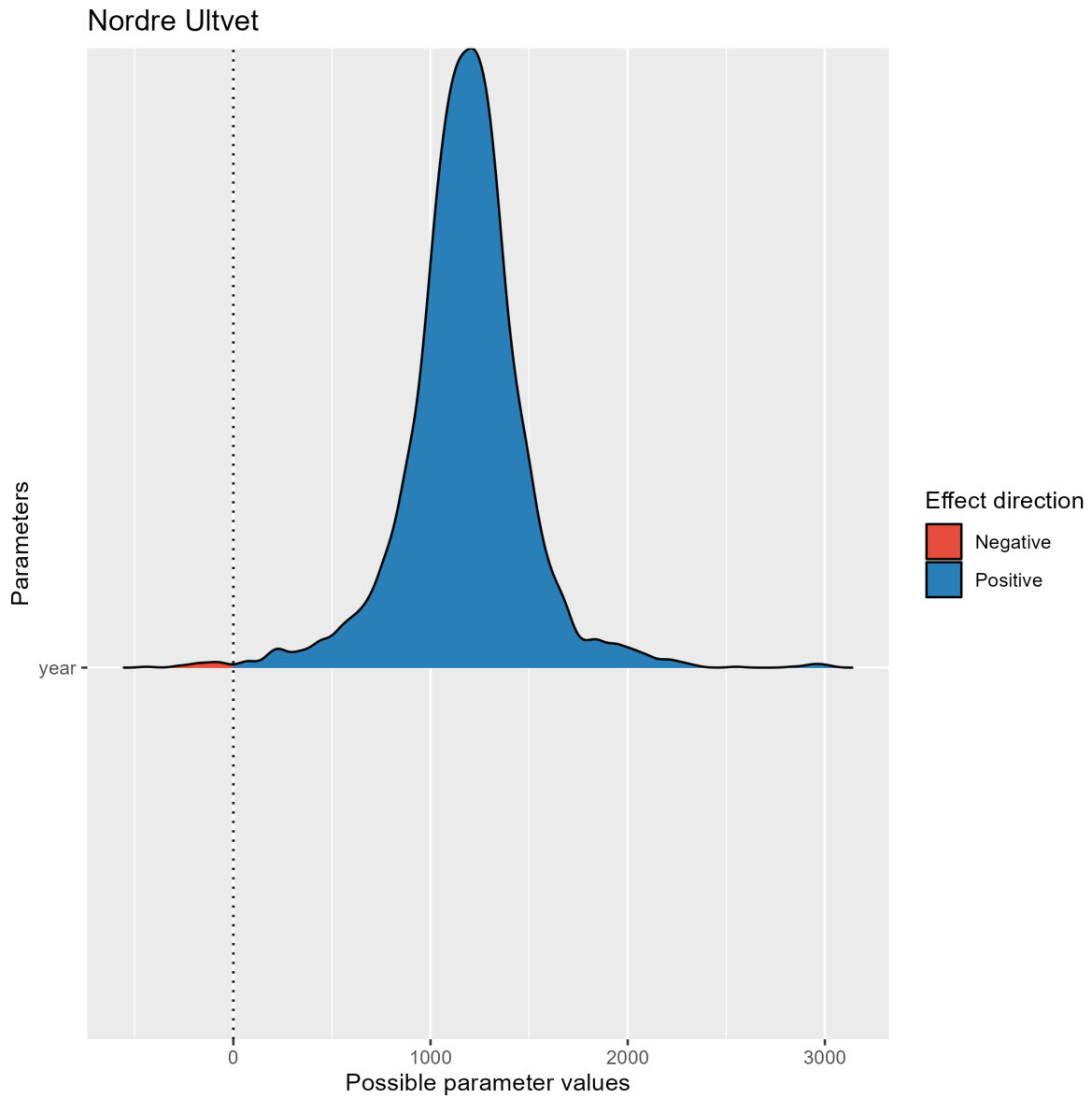


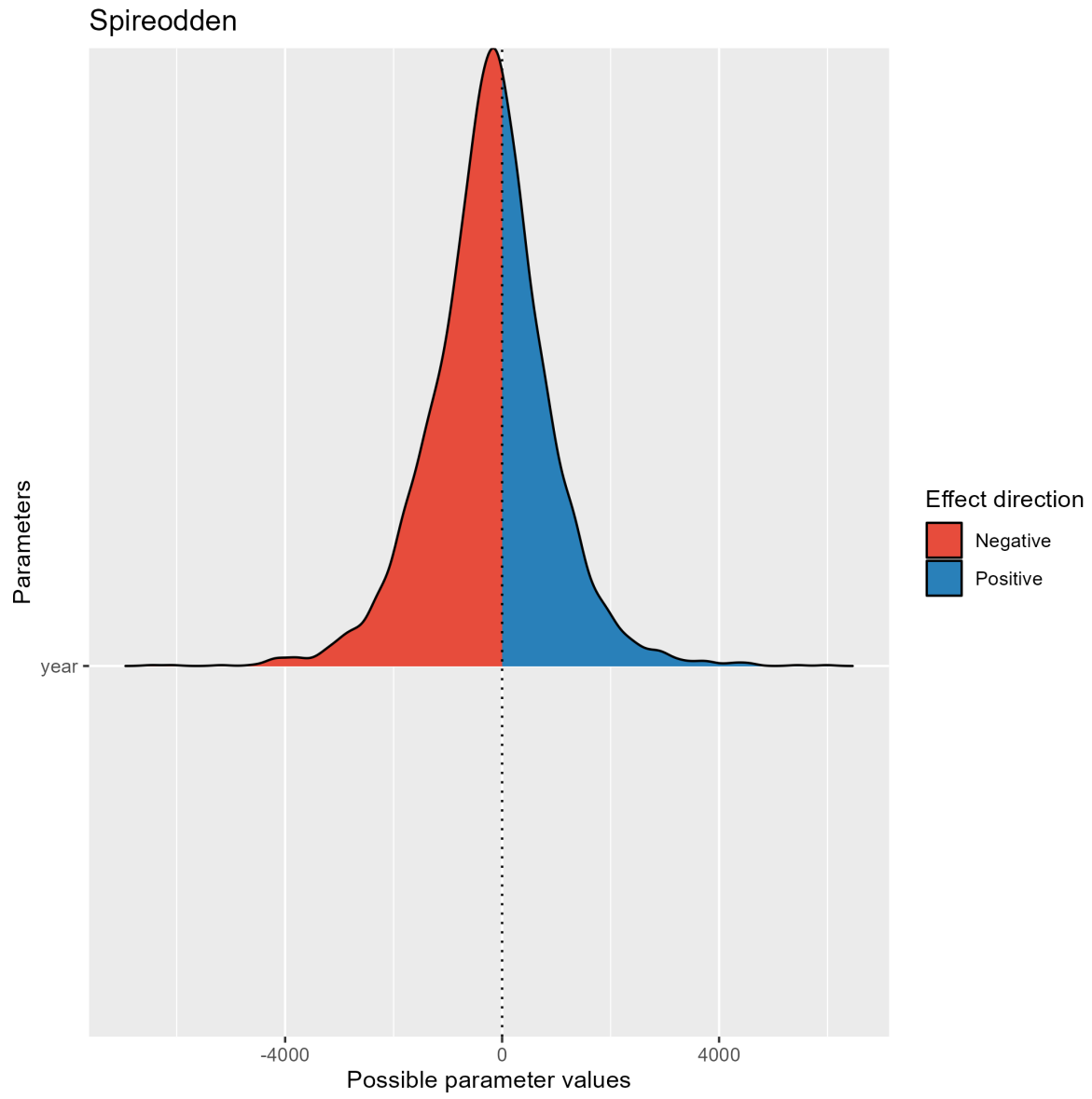


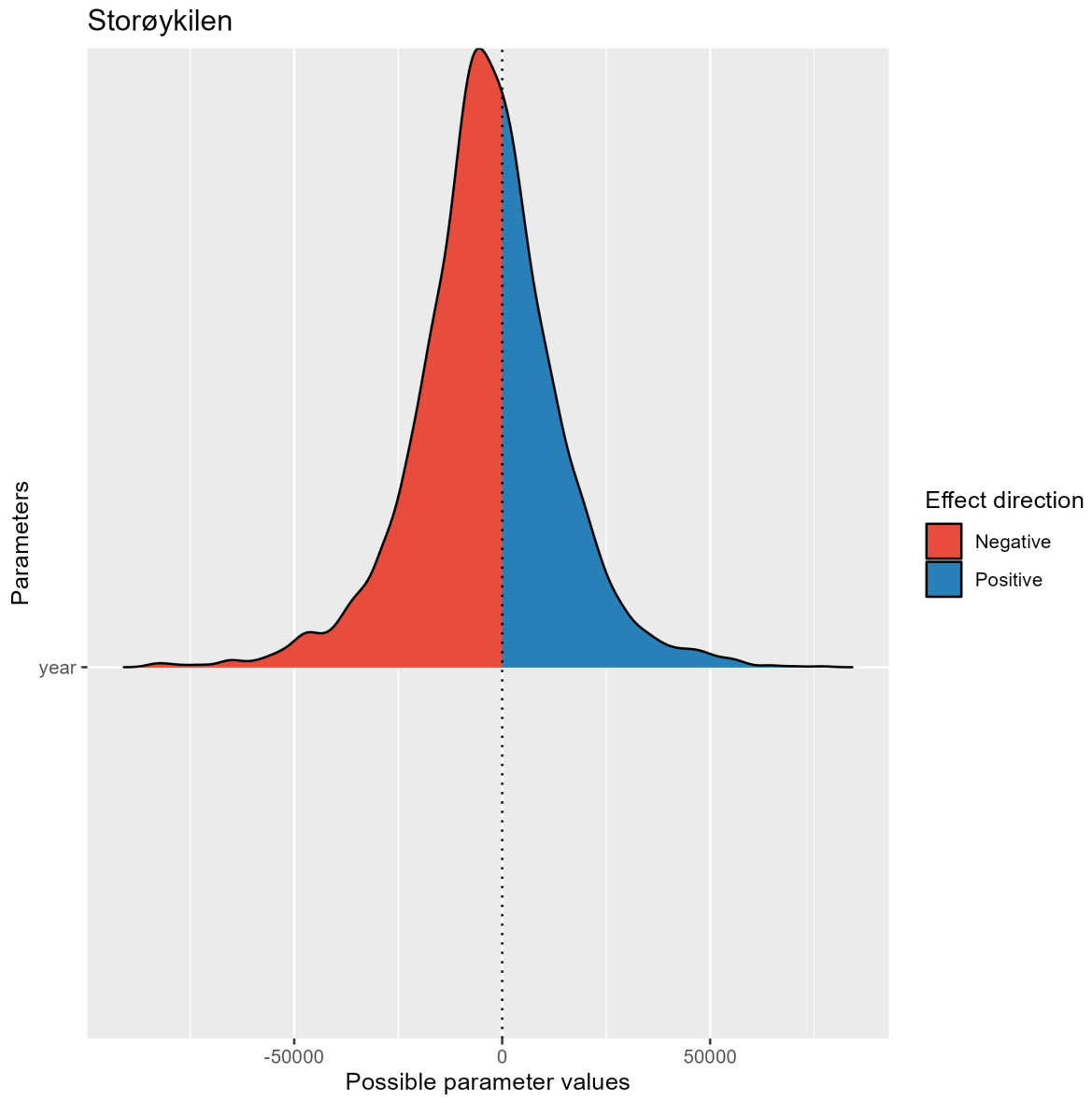


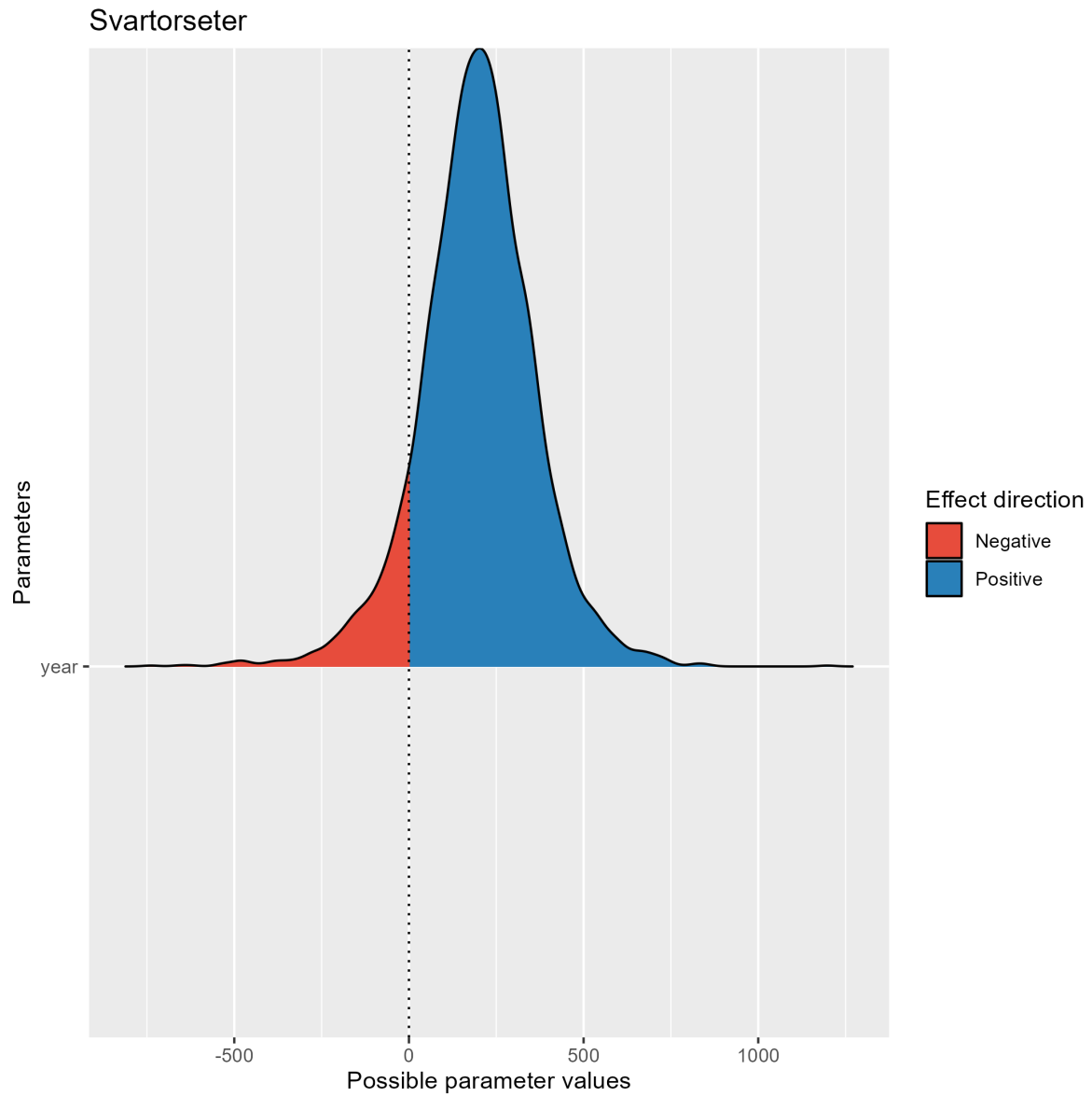




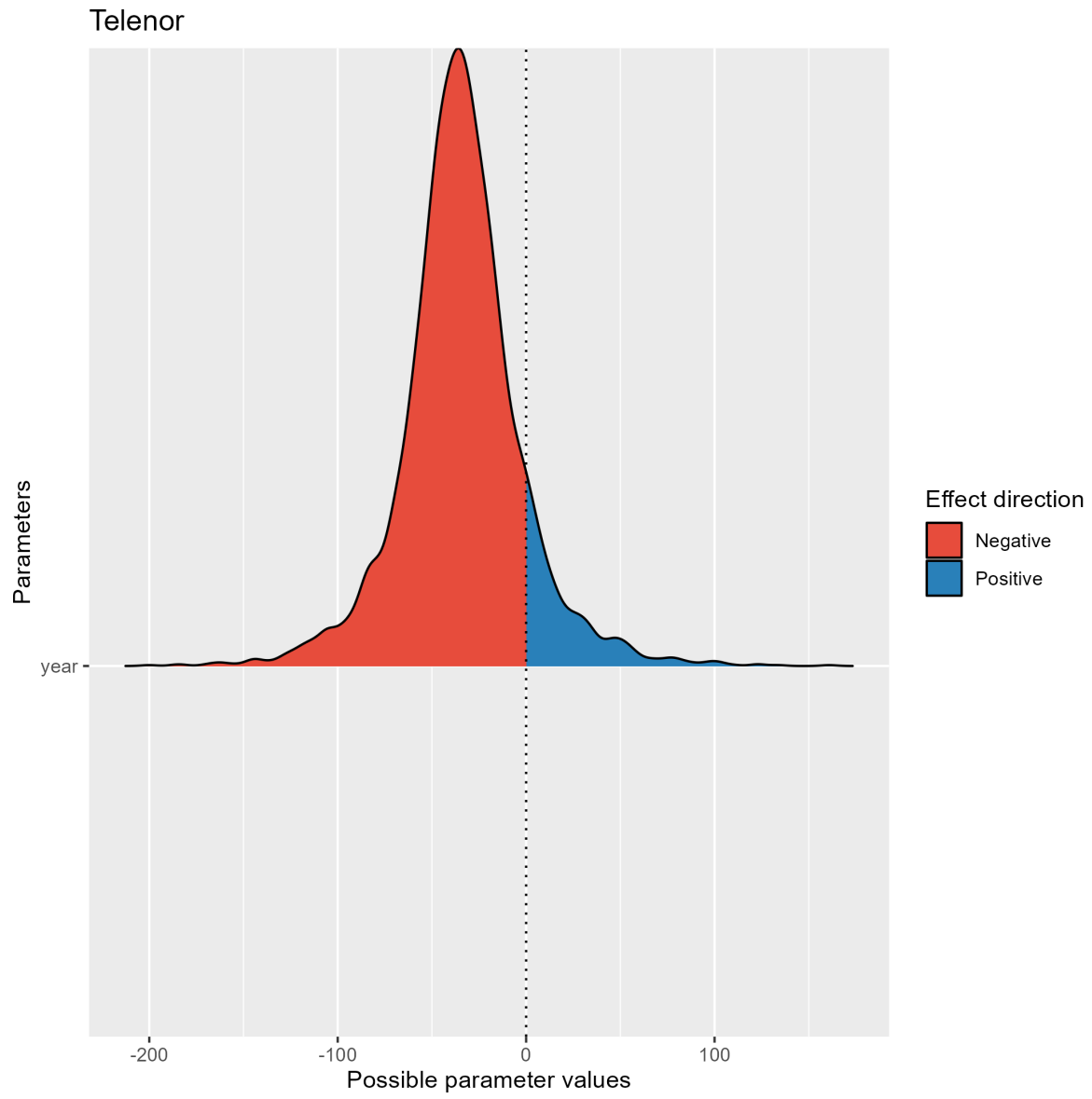


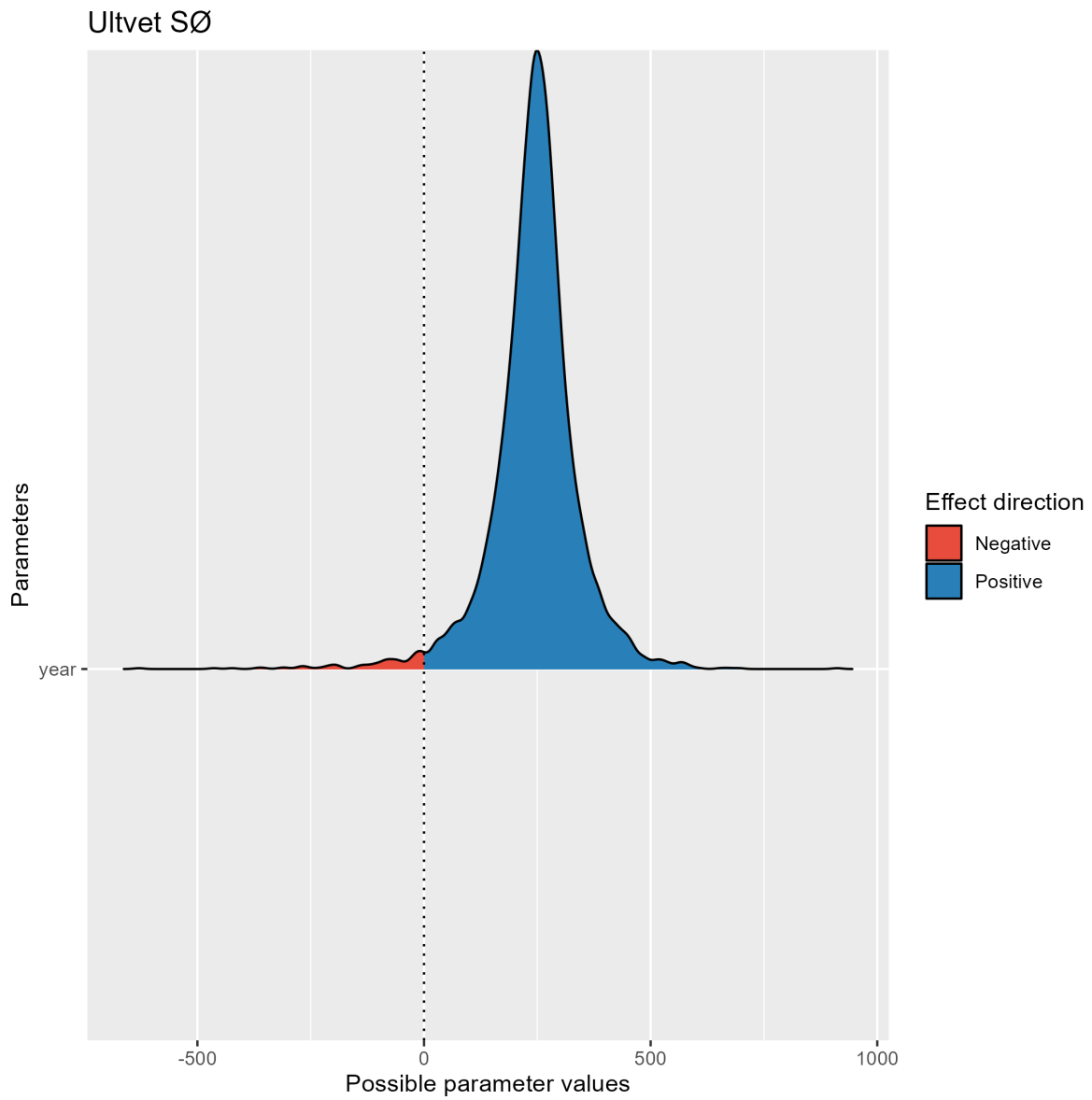


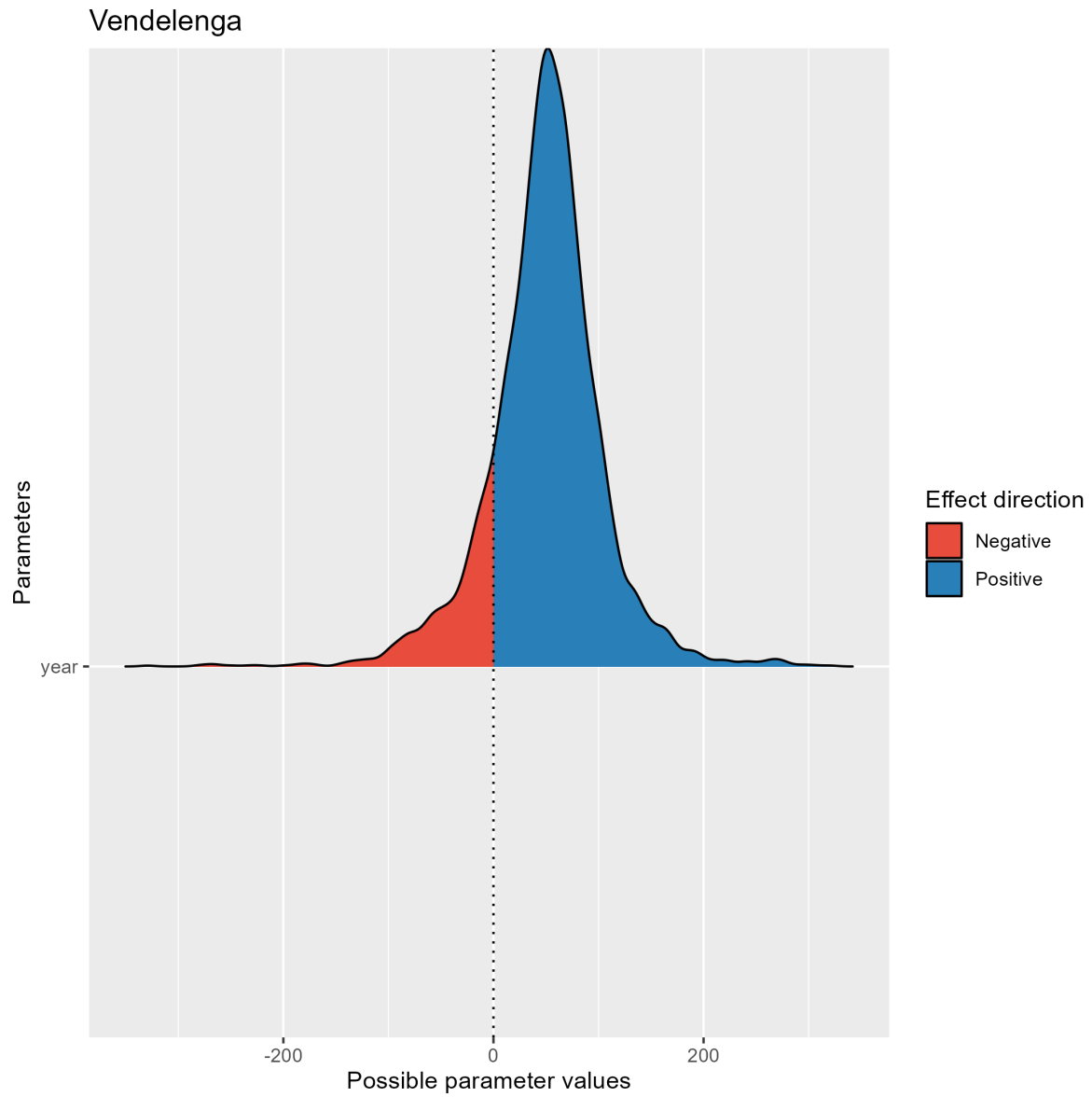


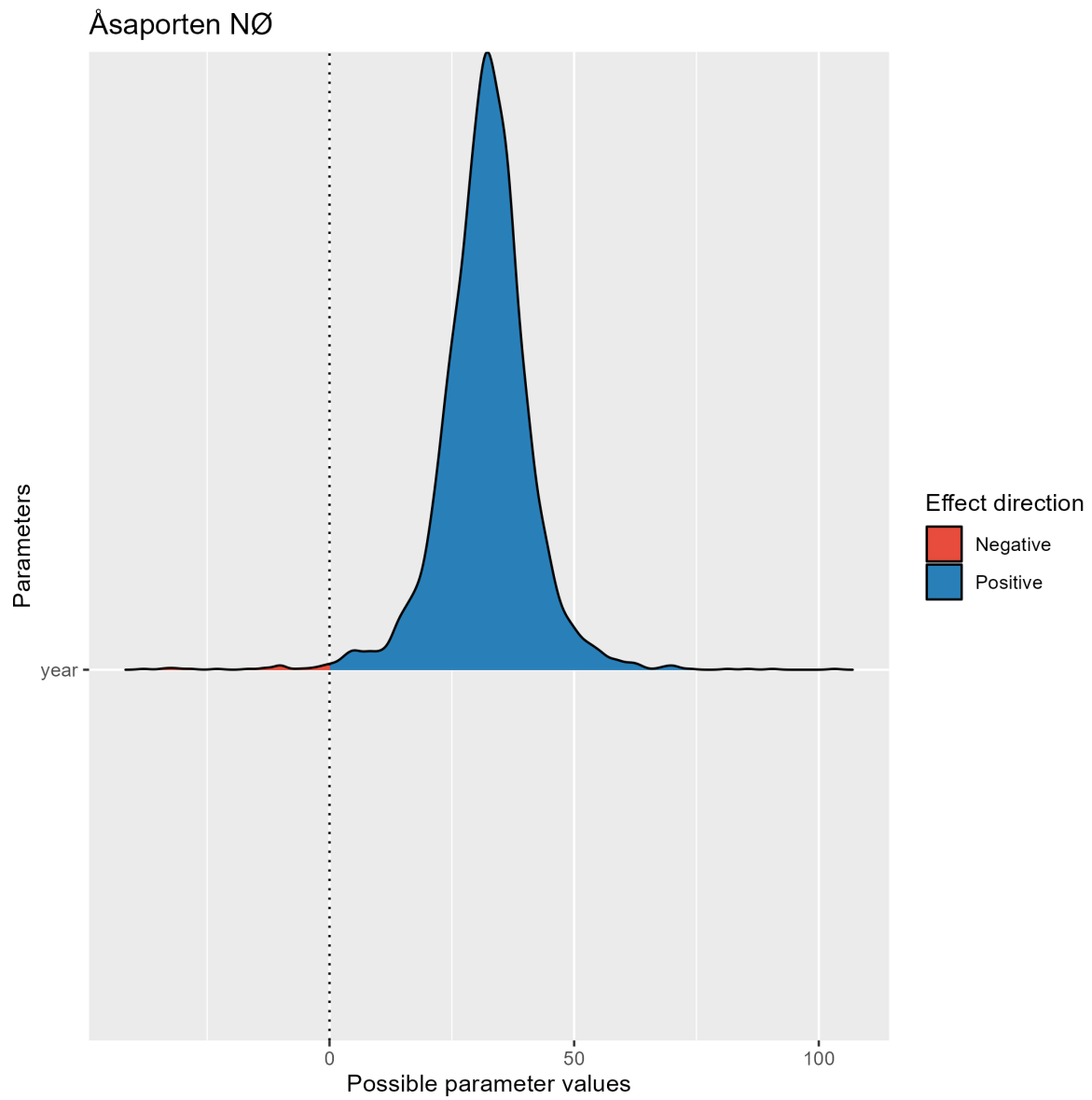












## 6 Code to recreate the analysis

To look at the functions used go to [Chapter 7](#)

```
# Population estimation script ----
## Load libraries ----
library(boot)
library(rio)
library(Dragehode)
library(tidyverse)
#Sys.setlocale(locale = 'Norwegian Bokmål_Norway')
# Uncomment this line if you locale is not Norway
source('R/funksjoner.R', encoding = "UTF-8")

## Dataimport ----

### save the data with the addition of this years data as "Dragehode_overvaking_DATA.xlsx"
### To reduce the burden (data storage on GitHub is not recommended) on the GitHub repo th

lokalitetsdata <- import("data/Dragehode_overvaking_DATA.xlsx",sheet="Lokalitetsdata",na=c("NA"))
transektdata <- import("data/Dragehode_overvaking_DATA.xlsx",sheet="Transektdata",na=c("NA"))
rutedata <- import("data/Dragehode_overvaking_DATA.xlsx",sheet="Rutedata",na=c("NA"))
artsdata <- import("data/Dragehode_overvaking_DATA.xlsx",sheet="Artsdata",na=c("NA"))
korreksjonsdata <- import("data/siriskuddogind.xlsx")

### Save the data that has been imported to overwrite last years files ----

saveRDS(lokalitetsdata, "data/lokalitetsdata.RDS")
saveRDS(transektdata, "data/transektdata.RDS")
saveRDS(rutedata, "data/rutedata.RDS")
saveRDS(artsdata, "data/artsdata.RDS")
saveRDS(korreksjonsdata, "data/korreksjonsdata.RDS")

# Datajusteringer ----
```

```
### Remove NAs ----
```

```
rutedata <- rutedata[!is.na(rutedata$Lokalitet), ]  
lokalitetsdata <- lokalitetsdata[!apply(lokalitetsdata,1,function(x){all(is.na(x))}),]  
transektdata <- transektdata[!apply(transektdata,1,function(x){all(is.na(x))}),]  
rutedata <- rutedata[!apply(rutedata,1,function(x){all(is.na(x))}),]
```

```
### change format of Lokalitet to factor ----
```

```
lokalitetsdata$Lokalitet <- factor(lokalitetsdata$Lokalitet)
```

```
### data to be omitted (single sampling event or change in method in earliest years) ----
```

```
lok.utelates <- c("Nedre Rykenvik", "vre Rykenvik")
```

```
lokalitetsdata <- lokalitetsdata[!(lokalitetsdata$Lokalitet%in%lok.utelates),]; lokalitetsdata
```

```
transektdata <- transektdata[!(transektdata$Lokalitet%in%lok.utelates),]; transektdata$Lokalitet
```

```
rutedata <- rutedata[!(rutedata$Lokalitet%in%lok.utelates),]; rutedata$Lokalitet <- factor(rutedata$Lokalitet)
```

```
lokalitetsdata <- lokalitetsdata[!(lokalitetsdata$Lokalitet%in%c("Falang") & lokalitetsdata$
```

```
transektdata <- transektdata[!(transektdata$Lokalitet%in%c("Falang") & transektdata$r<2020),]
```

```
rutedata <- rutedata[!(rutedata$Lokalitet%in%c("Falang") & rutedata$r<2020),]
```

```
### Create a hovednaturtype by combining the Naturtypes in to a single column ----
```

```
lokalitetsdata$Hovednaturtype <- sapply(strsplit(lokalitetsdata$Naturtype,"-"),function(x){
```

```
### Change character data to character formats and create a hovedskj tsel coloumn ----
```

```
lokalitetsdata$Hovedskj tsel <- as.character(lokalitetsdata$Krattrydding_bin r)
```

```
lokalitetsdata$Hovedskj tsel[lokalitetsdata$Krattrydding_bin r=="Ja"] <- "Krattrydding"
```

```
lokalitetsdata$Hovedskj tsel[lokalitetsdata$Sl tt_bin r=="Ja"] <- "Sl tt"
```

```
lokalitetsdata$Hovedskj tsel[lokalitetsdata$Hovedskj tsel=="Nei"] <- "Ingen/annet"
```

```
lokalitetsdata$Hovedskj tsel <- factor(lokalitetsdata$Hovedskj tsel)
```

```
### convert "ingen" to NA ----
```

```
transektdata$Forekomst dragehode (m)`[transektdata$Forekomst dragehode (m)`%in%c("ingen")
```

```
# Get coverage ----
```

```
artsdata$Vedplante<-artsdata$`Vedplante i feltsjikt (NY)`
```

```
rutedata$Vedpl.dekn <- beregn.dekning(rutedata,artsdata,"Vedplante")
```

```
lokalitetsnavn <- levels(lokalitetsdata$Lokalitet)
```

```
regionnavn <- levels(factor(lokalitetsdata$Region))
```

```
naturtypenavn <- levels(factor(lokalitetsdata$Hovednaturtype))
```

```

# Horgen og Mllerenga: tellinger av skudd (vegetative og fertile) korrigeres til estimert
names(korreksjonsdata)[6:11] <- c("x1","x2","x3","y1","y2","y3")
#par(mfrow=c(2,2))
#plot(korreksjonsdata$x1,korreksjonsdata$y1,main="sm planter")
m1 <- lm(y1~-1+x1,data=korreksjonsdata)
# summary(m1)
# abline(m1)
# plot(korreksjonsdata$x2,korreksjonsdata$y2,main="vegetative")
m2 <- lm(y2~-1+x2,data=korreksjonsdata)
# summary(m2)
# abline(m2)
#plot(korreksjonsdata$x3,korreksjonsdata$y3,main="fertile")
m3 <- lm(y3~-1+x3,data=korreksjonsdata)
# summary(m3)
# abline(m3)

i <- rutedata$Lokalitet%in%c("Horgen","Mllerenga") & rutedata$r%in%c(2019,2020)
newdat <- rutedata[i,c("Veg.planter","Fert.planter")]
names(newdat) <- c("x2","x3")
y2 <- round(predict(m2,newdat))
y3 <- round(predict(m3,newdat))
y4 <- rutedata[i,"Sm planter"]+y2+y3

#par(mfrow=c(1,3))
#plot(rutedata[i,"Veg.planter"],y2)
#plot(rutedata[i,"Fert.planter"],y3)
#plot(rutedata[i,"Ant.DR"],y4)

rutedata[i,"Veg.planter"] <- y2
rutedata[i,"Fert.planter"] <- y3
rutedata[i,"Ant.DR"] <- y4

# Populasjonsstrelser ----

# Sjekk om forekomst varierer med avstand fra midtpunkt
# par(mfrow=c(1,1))
# j <- rutedata$Lokalitet!="Ekebergskringen"
# plot(rutedata$Avst[j],rutedata$Ant.DR[j]>0,main="Alle lokaliteter")
# par(mfrow=c(5,5))
# for(i in 1:length(lokalitetsnavn))
# {

```

```

# j <- rutedata$Lokalitet==lokalitetsnavn[i]
# plot(rutedata$Avst[j],rutedata$Ant.DR[j]>0,main=lokalitetsnavn[i])
# }
#
# # Sjekk om tetthet varierer med avstand fra midtpunkt
# par(mfrow=c(1,1))
# j <- rutedata$Ant.DR>0 & rutedata$Lokalitet!="Ekebergskrinningen"
# plot(rutedata$Avst[j],rutedata$Ant.DR[j],main="Alle lokaliteter")
# par(mfrow=c(5,5))
# for(i in 1:length(lokalitetsnavn))
# {
#   j <- rutedata$Lokalitet==lokalitetsnavn[i] & rutedata$Ant.DR>0
#   plot(rutedata$Avst[j],rutedata$Ant.DR[j],main=lokalitetsnavn[i])
# }

# For hver populasjon (fordi de har sregenheter i design, og det er nskelig presentere
# Ogs for hvert r (Hensiktsmessig hvis f.eks. design endrer seg...

## lokalestimator ----
lokalitetsestimator <- list()
for(i in 1:length(lokalitetsnavn))
{
  tryCatch({
    print(lokalitetsnavn[i])
    popstr <- beregn.popstruktur(lokalitetsnavn[i],lokalitetsdata,transektdata,rutedata,qu
    lokalitetsestimator[[i]] <- popstr

  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}
names(lokalitetsestimator) <- lokalitetsnavn

#save the Lokalitetsestimator

saveRDS(lokalitetsestimator,"data/derived_data/lokalitetsestimator.RDS" )

### Figurer ----

if(!dir.exists("Figurer")) dir.create("Figurer")

# "R " figurer av alle lokaliteter med populasjonsestimator m/CI over tid for fertile, veg

```



```

#par(mfrow=c(5,5))
pdf(paste("Figurer/", "Alle lokaliteter", ".pdf", sep=""))
for(i in 1:length(lokalitetsestimater))
{
  tryCatch({
    popstr <- lokalitetsestimater[[i]]
    plot.popstruktur(popstr)

}, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})

}
dev.off()

for(i in 1:length(lokalitetsestimater))
{
  tryCatch({
    #pdf(paste("Figurer/", lokalitetsnavn[i], ".pdf", sep=""))
    #png(paste("Figurer/", lokalitetsnavn[i], ".png", sep=""))
    jpeg(paste("Figurer/", lokalitetsnavn[i], ".jpg", sep=""))
    popstr <- lokalitetsestimater[[i]]
    plot.popstruktur(popstr)
  }, error=function(e){cat("ERROR:", conditionMessage(e), "\n")})

  dev.off()
}

# Populasjonstrender per gruppe, totalt, fertile, vegetative og sm

lokfarge <- rainbow(length(lokalitetsnavn)); names(lokfarge) <- lokalitetsnavn
regsymbol <- c(1,2,3,4,5,6); names(regsymbol) <- regionnavn
natlinje <- c(1,2); names(natlinje) <- naturtypenavn

thisyear=lubridate::year(Sys.time())

tidsrom=c(2016.5, thisyear)

if(!dir.exists("Figurer/Regioner")) dir.create("Figurer/Regioner")
# Regioner
pdf(paste("Figurer/Regioner/", "Alle regioner", ".pdf", sep=""))
plot.gruppetrender(lokalitetsdata,gruppevariabel="Region",lokalitetsestimater,reverser=T,1

```

```

par(mfrow=c(1,3))
plot(0,0,type="n",axes=F,xlab="",ylab=""); legend("topleft",lty=1,col=lokfarge,legend=loka
dev.off()

if(!dir.exists("Figurer/Naturtyper")) dir.create("Figurer/Naturtyper")
# Naturtyper
pdf(paste("Figurer/Naturtyper/", "Alle hovedtyper", ".pdf", sep=""))
plot.gruppeterender(lokalitetsdata,gruppevariabel="Hovednaturtype",lokalitetsestimater,rever
par(mfrow=c(1,3))
plot(0,0,type="n",axes=F,xlab="",ylab=""); legend("topleft",lty=1,col=lokfarge,legend=loka
dev.off()

if(!dir.exists("Figurer/Skj tsel")) dir.create("Figurer/Skj tsel")
pdf(paste("Figurer/Skj tsel/", "Hovedskj tsel", ".pdf", sep=""))
plot.gruppeterender(lokalitetsdata,gruppevariabel="Hovedskj tsel",lokalitetsestimater,rever
par(mfrow=c(1,3))
plot(0,0,type="n",axes=F,xlab="",ylab=""); legend("topleft",lty=1,col=lokfarge,legend=loka
dev.off()

## GGplot versions of figures
if(!dir.exists("Figurer/ggplots")) dir.create("Figurer/ggplots")
for(i in 1:length(lokalitetsestimater)){
  tryCatch({

    popstr <- list(lokalitet=lokalitetsestimater[i][[1]]$lokalitet,year=lokalitetsestimater
      nVeg=lokalitetsestimater[i][[1]]$nVeg,nSma=lokalitetsestimater[i][[1]]$
      Veg.CI=lokalitetsestimater[i][[1]]$Veg.CI,Sma.CI=lokalitetsestimater[i]

    P=ggplot.popstruktur(popstr)

  })
}

# # Populasjonsestimater 2017-2020

```

```

# utvalgte.lok <- lokalitetsdata$Lokalitet[lokalitetsdata$r==2017]
# for(i in 1:length(utvalgte.lok))
# {
#   lok <- utvalgte.lok[i]
#   print(lokalitetsestimater[lok])
# }

# Forekomst frav r?

# Vekstrater-----

popvekstdata <- beregn.vekstrate(rutedata,"Ant.DR","RuteID")
popvekstdata$logR <- log(popvekstdata$vekstrate)
popvekstdata$logR[popvekstdata$logR%in%c(-Inf,Inf)] <- NA
#summary(popvekstdata$logR)
i <- match(popvekstdata$Lokalitet,lokalitetsdata$Lokalitet)
popvekstdata$Region <- lokalitetsdata$Region[i]
popvekstdata$Hovednaturtype <- lokalitetsdata$Hovednaturtype[i]
popvekstdata$Skj tsel_bin r <- lokalitetsdata$Skj tsel_bin r[i]
popvekstdata <- popvekstdata[!is.na(popvekstdata$Veg.h yde),]

## save popvekstdata
saveRDS(popvekstdata, "data/derived_data/popvekstdata.RDS")

# par(mfrow=c(1,1))
# plot(popvekstdata[,c("vekstrate","logR","Ant.DR","Bunnsjikt","Feltsjikt","Busksjikt","Tr
# plot(popvekstdata[popvekstdata$r==2017,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])
# plot(popvekstdata[popvekstdata$r==2018,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])
# plot(popvekstdata[popvekstdata$r==2019,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])
# plot(popvekstdata[popvekstdata$r==2020,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])
# plot(popvekstdata[popvekstdata$r==2021,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])
# plot(popvekstdata[popvekstdata$r==2022,c("vekstrate","logR","Veg.h yde","Vedpl.dekn")])

# library(gamm4)
#
# popvekstdata$logVH <- log(popvekstdata$Veg.h yde)
# popvekstdata$Hovednaturtype <- factor(popvekstdata$Hovednaturtype)

```

```

# popvekstdata$Year <- factor(popvekstdata$r)
#
# m1.gam <- gamm4(logR~Hovednaturtype+s(Veg.h yde,by=Hovednaturtype),random=~(1|Region)+(1|Lokalitet),data=popvekstdata)
# summary(m1.gam$gam)
# ranef(m1.gam$mer)
#
# m1.gam.b <- gamm4(logR~Hovednaturtype+s(Veg.h yde,by=Hovednaturtype)+s(Vedpl.dekn),random=~(1|Region)+(1|Lokalitet),data=popvekstdata)
# summary(m1.gam.b$gam)
#
# m2.gam <- gamm4(logR~Hovednaturtype+s(logVH,by=Hovednaturtype),random=~(1|Region)+(1|Lokalitet),data=popvekstdata)
# summary(m2.gam$gam)
#
#
# yr <- seq(from=min(popvekstdata$r), to=max(popvekstdata$r), by=1)
# modlist <- vector("list",length(yr)); names(modlist) <- yr
# for(i in 1:length(yr))
# {
#   y <- popvekstdata$Year==yr[i] # Bare en Region (Oslo) i 2017, s ingen random effect for lokalitet
#   if(yr[i]==2017)
#   {
#     m1 <- gamm4(logR~Hovednaturtype+s(Veg.h yde,by=Hovednaturtype),random=~(1|Lokalitet),data=popvekstdata[y,])
#     m2 <- gamm4(logR~Hovednaturtype+s(logVH,by=Hovednaturtype),random=~(1|Lokalitet),data=popvekstdata[y,])
#   }
#   else
#   {
#     m1 <- gamm4(logR~Hovednaturtype+s(Veg.h yde,by=Hovednaturtype),random=~(1|Region)+(1|Lokalitet),data=popvekstdata)
#     m2 <- gamm4(logR~Hovednaturtype+s(logVH,by=Hovednaturtype),random=~(1|Region)+(1|Lokalitet),data=popvekstdata)
#   }
#   modlist[[i]] <- list(m1.gam=m1,m2.gam=m2)
# }
#
# pdf(paste("Figurer/Vekstrater/", "Alle plot", ".pdf", sep=""))
# par(mfrow=c(2,2),mar=c(4,4,2,0.1))
# xlim <- c(0,max(popvekstdata$Veg.h yde[!is.na(popvekstdata$logR)],na.rm=T))
# ylim <- range(popvekstdata$logR,na.rm=T)
# xlab <- c("", "Vegetasjonsh yde (cm)", "Vegetasjonsh yde (cm)")
# ylab <- c("", "log R", "")
# main <- c("2017-2018", "2018-2019", "2019-2020")
#
# i <- as.numeric(popvekstdata$Hovednaturtype=="T2")+1
# plot(popvekstdata$Veg.h yde, popvekstdata$logR, pch=21, cex=0.8, bg=c("blue", "red")[i], col=c("black", "black")[i])

```

```

# lines(popvekstdata$Veg.h yde,predict(m1.gam$gam,newdata=data.frame(Veg.h yde=popvekstdata
# lines(popvekstdata$Veg.h yde,predict(m1.gam$gam,newdata=data.frame(Veg.h yde=popvekstdata
# abline(h=0,lty=3)
# legend("bottomright",c("T32 Seminaturalig eng","T2 pen grunnlendt mark"),lty=c(1,1),col=
#
# for(i in 1:length(yr))
# {
#   y <- popvekstdata$Year==yr[i]
#   j <- as.numeric(popvekstdata$Hovednaturtype[y]=="T2")+1
#   plot(popvekstdata$Veg.h yde[y],popvekstdata$logR[y],pch=21,cex=0.8,bg=c("blue","red")[
#
#   if(i>1)
#   {
#     VH <- popvekstdata$Veg.h yde[y][!is.na(popvekstdata$logR[y])&popvekstdata$Hovednatur
#     Veg.h yde <- min(VH,na.rm=T):max(VH,na.rm=T)
#     lines(Veg.h yde,predict(modlist[[i]]$m1.gam$gam,newdata=data.frame(Veg.h yde=Veg.h yd
#   }
#   VH <- popvekstdata$Veg.h yde[y][!is.na(popvekstdata$logR[y])&popvekstdata$Hovednaturty
#   Veg.h yde <- min(VH,na.rm=T):max(VH,na.rm=T)
#   lines(Veg.h yde,predict(modlist[[i]]$m1.gam$gam,newdata=data.frame(Veg.h yde=Veg.h yde,l
#
#   abline(h=0,lty=3)
#   legend("bottomright",c("T32 Seminaturalig eng","T2 pen grunnlendt mark"),lty=c(1,1),col
# }
# dev.off()
#
#
# # Moodellsammendrag, tabell av random effects
# summary(m1.gam$gam)
# summary(m1.gam$mer)
# ranef(m1.gam$mer)
#
# coef(m1.gam$mer)
#
# summary(modlist[[3]]$m1.gam$gam)
#
# table(popvekstdata$r, popvekstdata$Hovednaturtype, popvekstdata$logR>0)

```

## 7 Funksjoner

To see the code which uses these functions go to [Chapter 6](#)

```
#' Calculate coverage
#'
#' @param rutedata quadrature data
#' @param artsdata species data
#' @param artsgruppe Species
#
#' @return coverage
#' @export
#
beregndekning <- function(rutedata,artsdata,artsgruppe){
  dekning <- rep(0,nrow(rutedata))
  for(i in 1:nrow(rutedata))
  {
    utvalgte.data <- artsdata[,artsgruppe]==1 & artsdata$Rute==rutedata$RuteID[i] & artsda
    if(length(utvalgte.data)>0) dekning[i] <- sum(artsdata$Dekning[utvalgte.data],na.rm=T)
  }
  return(dekning)
}

#
#
#' Calculate area weight
#
#' @param x Quadrature data
#' @param design Linje or Rose
#' @param lokalitetsbredde Width of locality
#' @param rutebredde width of quadrature (defaults to 1)
#
#' @return area weighting of survey to calculate estimates
#' @keywords internal
#
beregnaarealvekt <- function(x,design,lokalitetsbredde=NA,rutebredde=1){
```

```

if(design=="Linje")
{
  return(rep(lokalitetsbredde/rutebredde,length(x)))
}
if(design=="Rose")
{
  r1 <- x          # radius for indre sirkel (rutas avstand fra startpunktet)
  r2 <- x+rutebredde # radius for ytre sirkel
  return( (pi*r2^2 - pi*r1^2)/8 )
}
}

#' Bootstrap population
#'
#' @param tetthet density
#' @param forekomstareal Occurrence area
#' @param nboot number of bootstrap iterations (defaults to 2000)
#'
#' @return bootstrapped estimates of population per age stage
#' @keywords internal
#'
boot.pop <- function(tetthet,forekomstareal,nboot=2000){
  tetthet <- tetthet[!is.na(tetthet)]
  if(length(tetthet)==0) {print("Ingen individer"); return(rep(NA,nboot))}
  tetthet.boot <- boot(tetthet,statistic=function(x,k){mean(x[k])},R=nboot)
  forekomstareal.boot <- boot(forekomstareal,statistic=function(x,k){sum(x[k])},R=nboot)
  tetthet.boot$t * forekomstareal.boot$t
}

#' Calculate the Occurrence area
#'
#' @param transektdata transect data
#' @param utvalgt.lokalitet chosen locality
#' @param year year
#' @param design Linje or Rose
#' @param lokalitetsbredde Width of locality
#' @param rutebredde width of quadrat (defaults to 1)
#' @param forekomst_transekt "Dragehode" column in data

```

```

#' @param forekomst_avstand "Forekomst dragehode (m)" column in data
#'
#' @keywords internal
beregns.forekomstareal <- function(transektdata,utvalgt.lokalitet,year,design,lokalitetsbre
                                forekomst_transekt="Dragehode",forekomst_avstand="Foreko

  x <- list()
  for(i in 1:length(year))
  {
    transekter <- transektdata[transektdata$Lokalitet==utvalgt.lokalitet & transektdata$r

    if(all(is.na(transekter[,forekomst_transekt])))
    {
      next
    }
    if(all(transekter[,forekomst_transekt]==0))
    {
      next
    }
    forekomststruter <- as.numeric(unlist(sapply(transekter[,forekomst_avstand],strsplit,spl
    forekomststruter <- forekomststruter[!is.na(forekomststruter)]
    forekomstareal <- beregn.arealvekt(forekomststruter,design[i],lokalitetsbredde[i],rutebr
    x[[i]] <- forekomstareal
  }
  mangler <- unlist(lapply(x,is.null))
  if(any(mangler))
  {
    erstattes <- which(mangler)
    registrert <- which(!mangler)
    #print(erstattes)
    #print(registrert)
    for(i in erstattes)
    {
      delta <- abs((registrert+0.5) - i)
      #print(delta)
      j <- registrert[delta==min(delta)]
      #print(j)
      x[[i]] <- x[[j]]
    }
  }
  return(x)

```



```

}

#' Calculate population structure
#'
#' @param utvalgt.lokalitet chosen locality
#' @param lokalitetsdata data for that locality
#' @param transektdata transect data
#' @param rutedata quadrate data
#' @param rutebredde quadrate width (defaults to 1)
#' @param fert Fertile plants column
#' @param veg Vegetative plants column
#' @param sma seedlins column
#' @param tot total plants column
#' @param forekomst_transekt occurrence transect data
#' @param forekomst_avstand occurrence distance
#' @param quantiles quantiles to be calculated (defaults to 0.025, 0.975)
#'
#' @return population structure list per location
#' @export
#'

beregnpopstruktur <- function(utvalgt.lokalitet,lokalitetsdata,transektdata,rutedata,rutebredde,fert="Fert.planter",veg="Veg.planter",sma="Småplanter",tot="Totalplanter",forekomst_transekt="Dragehode",forekomst_avstand="Forekomst",quantiles=c(0.025,0.975)){

  year <- lokalitetsdata[lokalitetsdata$Lokalitet==utvalgt.lokalitet,"r"]
  design <- lokalitetsdata[lokalitetsdata$Lokalitet==utvalgt.lokalitet,"Design"]
  lokalitetsbredde <- lokalitetsdata[lokalitetsdata$Lokalitet==utvalgt.lokalitet,"Lokalite
  print(paste(utvalgt.lokalitet, year, design,"bredde:",lokalitetsbredde))
  nboot <- 2000
  nq <- length(quantiles)
  nyear <- length(year)
  nFert <- numeric(nyear)
  nVeg <- numeric(nyear)
  nSma <- numeric(nyear)
  nTot <- numeric(nyear)
  Fert.CI <- matrix(NA,nyear,nq)
  Veg.CI <- matrix(NA,nyear,nq)
  Sma.CI <- matrix(NA,nyear,nq)
  Tot.CI <- matrix(NA,nyear,nq)

```

```

forekomstareal.liste <- beregn.forekomstareal(transektdata,utvalgt.lokalitet,year,design)
#print(forekomstareal.liste)

for(i in 1:length(year))
{
  forekomstareal <- forekomstareal.liste[[i]]
  saveRDS(forekomstareal, paste0("data/derived_data/", utvalgt.lokalitet,"_forekomstareal",i,".RDS"))
  print(year[i])
  ruter <- rutedata[rutedata$Lokalitet==utvalgt.lokalitet & rutedata$r==year[i] & rutedata$Design==design]
  saveRDS(ruter, paste0("data/derived_data/", utvalgt.lokalitet,"_ruter","_",i,".RDS"))

  print("Fertile")
  if(!is.na(fert)) x <- boot.pop(ruter[,fert],forekomstareal,nboot)
  else x <- NA
  #print(summary(x))
  nFert[i] <- mean(x); Fert.CI[i,] <- quantile(x,quantiles,na.rm=T)

  print("Vegetative")
  if(!is.na(veg)) x <- boot.pop(ruter[,veg],forekomstareal,nboot)
  else x <- NA
  #print(summary(x))
  nVeg[i] <- mean(x); Veg.CI[i,] <- quantile(x,quantiles,na.rm=T)

  print("Småplanter")
  if(!is.na(sma)) x <- boot.pop(ruter[,sma],forekomstareal,nboot)
  else x <- NA
  #print(summary(x))
  nSma[i] <- mean(x); Sma.CI[i,] <- quantile(x,quantiles,na.rm=T)

  print("Totalt")
  if(!is.na(tot)) x <- boot.pop(ruter[,tot],forekomstareal,nboot)
  else x <- NA
  print(summary(x))
  nTot[i] <- mean(x); Tot.CI[i,] <- quantile(x,quantiles,na.rm=T)
}

popstr <- list(lokalitet=utvalgt.lokalitet,year=year,nFert=nFert,nVeg=nVeg,nSma=nSma,nTot=nTot)
return(popstr)
}

```

```

#' Base plot population structure
#'
#' @param popstr population structure data
#'
#' @return Baseplots of population structure per location
#' @export

plot.popstruktur <- function(popstr){
  thisyear=lubridate::year(Sys.time())
  tidsrom=c(2016.5, thisyear)
  plot(popstr$year,popstr$nTot,ylim=c(0,max(popstr$Tot.CI,na.rm=T)),type="o",main=popstr$year)
  polygon(c(popstr$year,rev(popstr$year)),c(popstr$Tot.CI[,1],rev(popstr$Tot.CI[,2])),col="black",lty=1)
  lines(popstr$year,popstr$nFert,col="red",type="o"); polygon(c(popstr$year,rev(popstr$year)),c(popstr$nFert,rev(popstr$nFert)),col="red",lty=1)
  lines(popstr$year,popstr$nVeg,col="green",type="o"); polygon(c(popstr$year,rev(popstr$year)),c(popstr$nVeg,rev(popstr$nVeg)),col="green",lty=1)
  lines(popstr$year,popstr$nSma,col="blue",type="o"); polygon(c(popstr$year,rev(popstr$year)),c(popstr$nSma,rev(popstr$nSma)),col="blue",lty=1)
  #lines(popstr$year,popstr$nSma+popstr$nVeg+popstr$nFert,lty=2,type="o")
  legend("topleft",c("Totalt","Fertile","Vegetative","Småplanter"),lty=c(1,1,1,1),pch=c(1,1,1,1))
}

#' Base plot group trends
#'
#' @param lokalitetsdata locality data
#' @param gruppevariabel grouping variable
#' @param lokalitetsestimater local estimates data
#' @param reverser Reverse the axis? default is FALSE
#' @param lokalitetsnavn locality name
#' @param lokfarge locality colour
#' @param regsymbol Regional symbol
#' @param natlinje linetype for naturtype
#'
#' @return baseplots of group trends
#' @export
#'

plot.gruppetreder <- function(lokalitetsdata,gruppevariabel,lokalitetsestimater,
                             reverser=FALSE,lokalitetsnavn,lokfarge,regsymbol,natlinje){
  thisyear=lubridate::year(Sys.time())
  tidsrom=c(2016.5, thisyear)

  gruppe <- unique(lokalitetsdata[,gruppevariabel])
  if(reverser) gruppe <- rev(gruppe)

```

```

ngrp <- length(gruppe)
par(mfcol=c(4,ngrp),mar=c(2,4,2,0.2))
for(i in 1:ngrp)
{
  grp <- as.character(gruppe[i])
  print(grp)
  lok <- as.character(unique(lokalitetsdata[lokalitetsdata[,gruppevariabel]==grp,"Lokalitet"]))
  farge <- lokfarge[lok]
  #print(lok)
  #print(farge)

  variabler <- c("nTot","nFert","nVeg","nSma")
  if(i==1) ylab <- c("Antall totalt","Antall fertile","Antall vegetative","Antall småplanter")
  else ylab <- rep("",length(variabler))
  xlab <- c(rep("",length(variabler)-1),"År")
  main <- c(grp,rep("",length(variabler)-1))
  for(i in 1:length(variabler))
  {
    noplot <- TRUE
    for(j in 1:length(lokalitet))
    {
      #print(lokalitet[j])
      reg <- as.character(unique(lokalitetsdata[lokalitetsdata$Lokalitet==lok,"Region"]))
      symbol <- regsymbol[reg]
      nat <- as.character(unique(lokalitetsdata[lokalitetsdata$Lokalitet==lok,"Hovednaturtype"]))
      linje <- natlinje[nat]
      #print(linje)

      popstr <- lokalitetsestimater[lok[j]][[1]]
      if(length(popstr)==1) {print("Ingen estimator"); next}
      if(noplot)
      {
        plot(popstr$year,popstr[variabler[i]][[1]],xlim=tidsrom,ylim=c(1,70000),type="o",
             main=main[i],xlab=xlab[i],ylab=ylab[i],log="y")
        noplot <- FALSE
      }
      else lines(popstr$year,popstr[variabler[i]][[1]],type="o",col=farge[j]) #,pch=symbol[j]
    }
  }
}
}

```

```

#

#' Calculate growth rate
#'
#' @param rutedata quadrate data
#' @param popvar population variable (Total Dragehode )
#' @param idvar Id of the quadrate (RuteID)
#'
#' @return growthrate estimates
#' @export

beregnav.vekstrate <- function(rutedata,popvar,idvar){
  vekstrate <- rep(NA,nrow(rutedata))
  year <- sort(as.numeric(unique(rutedata$r)))
  nyear <- length(year)
  if(nyear<2) return(NA)
  for(i in 1:(nyear-1))
  {
    t1 <- rutedata$r==year[i]
    t2 <- rutedata$r==year[i+1]
    idmatch <- match(rutedata[t1,idvar],rutedata[t2,idvar])
    vekstrate[t1] <- rutedata[t2,popvar][idmatch] / rutedata[t1,popvar]
  }
  rutedata$vekstrate <- vekstrate
  return(rutedata)
}

#' GGplot version of plot.popstruktur
#'
#' @param popstr population structure
#'
#' @return plots of population structure
#' @export
#'

ggplot.popstruktur <- function(popstr){
  require(tidyverse)
  title=popstr$lokalitet
  popstr=as_tibble(popstr)
  names(popstr)<-c( "lokalitet" ,"year",          "Fertile",          "Vegetative",          "Småplanter")

```

```

      "Fert.CI",    "Veg.CI",    "Sma.CI",    "Tot.CI" )

popstrCI=popstr %>%
  mutate("Fert.CI_upper"= Fert.CI[,2]) %>%
  mutate("Fert.CI_lower"= Fert.CI[,1]) %>%
  mutate("Tot.CI_upper"= Tot.CI[,2]) %>%
  mutate("Tot.CI_lower"= Tot.CI[,1]) %>%
  mutate("Veg.CI_upper"= Veg.CI[,2]) %>%
  mutate("Veg.CI_lower"= Veg.CI[,1]) %>%
  mutate("Sma.CI_upper"= Sma.CI[,2]) %>%
  mutate("Sma.CI_lower"= Sma.CI[,1]) %>%
  select(!lokalitet) %>%
  pivot_longer(!year, names_to = "key", values_to = "value") %>%
  filter(key%in% c("Fert.CI", "Veg.CI", "Sma.CI", "Tot.CI")) %>%
  mutate(var=gsub(".CI","", key))

popstrCI=popstrCI%>%
  mutate(var=rep(c("Fertile", "Vegetative", "Småplanter", "Totalt"),(dim(popstrCI)[1]/4))
  mutate(upper=value[,2]) %>%
  mutate(lower=value[,1]) %>%
  select(!value)

popstr_plot=popstr %>%
  mutate("Fert.CI_upper"= Fert.CI[,2]) %>%
  mutate("Fert.CI_lower"= Fert.CI[,1]) %>%
  mutate("Tot.CI_upper"= Tot.CI[,2]) %>%
  mutate("Tot.CI_lower"= Tot.CI[,1]) %>%
  mutate("Veg.CI_upper"= Veg.CI[,2]) %>%
  mutate("Veg.CI_lower"= Veg.CI[,1]) %>%
  mutate("Sma.CI_upper"= Sma.CI[,2]) %>%
  mutate("Sma.CI_lower"= Sma.CI[,1]) %>%
  select(!lokalitet) %>%
  pivot_longer(!year, names_to = "key", values_to = "value") %>%
  #filter(!key%in% c("Fert.CI", "Veg.CI", "Sma.CI", "Tot.CI")) %>%
  filter(key%in% c("Fertile", "Småplanter", "Totalt", "Vegetative")) %>%
  mutate(upper=popstrCI$upper) %>%
  mutate(lower=popstrCI$lower)

```

```

p=popstr_plot%>%
  ggplot(aes(year,value[,1], colour=key))+
  geom_point(size=2)+
  geom_ribbon(aes(ymin=lower, ymax=upper, fill=popstr_plot$key), alpha=0.2)+
  scale_fill_manual(values = c("darkred", "darkblue", "darkgrey", "darkgreen"), guide="n
  geom_line( size=1.2)+
  scale_colour_manual(values = c("darkred", "darkblue", "darkgrey", "darkgreen"))+
  labs(x="Å r", y= "Antall individer")+
  ggtitle(title)+
  theme_classic()

ggsave(
  paste0(here::here(),"/Figurer/ggplots/" ,title, ".png"))
}

```