Analysis Code

# Libraries

```{r, message = F}

library(tidyverse)

library(lubridate)

library(comorbidity)

library(cowplot)

library(janitor)

library(vroom)

library(here)

options(scipen = 999)

theme\_set(theme\_cowplot())

```

# Shielding list

```{r message = F}

#load and tidy shielding list

#this file has individuals listed one time \*for each reason shielding\*

shielding\_original <-

 vroom(here("original\_data", 'Dash416\_Shield20201012\_Release.csv'),

 delim = "¬") %>%

 clean\_names() %>%

 rename(id = dash416\_release\_uid) %>%

 mutate(date\_addition =

 as\_date(ymd\_hms(earliest\_addition\_this\_chi)),

 date\_removal =

 as\_date(ymd\_hms(most\_recent\_removal\_date)),

 date\_death = as\_date(ymd(trak\_person\_deceased\_date))) %>%

 select(id,

 group,

 origin\_for\_this\_group,

 date\_addition,

 date\_removal,

 removal\_description,

 date\_death) %>%

 distinct()

#add translation of shielding group codes

shielding\_original <-

 vroom(here("original\_data", "shielding\_codes\_short.csv"),

 delim = ",") %>%

 left\_join(shielding\_original, ., by = c("group" = "shielding\_group"))

```

# Population exclusions

```{r}

#anyone who died before 26 March (first day of shielding)

#false positives from a lung cancer screen

#moved from Scotland

shielding\_long <-

 shielding\_original %>%

 filter(date\_death > ymd("2020-03-26") |

 is.na(date\_death)) %>%

 filter(

 !removal\_description %in%

 c("FalsePosLungCancer", "False Positive", "Moved out of Scotland"))

#vector of ids of people in final cohort for paper

unique\_ids <- unique(shielding\_long$id)

n\_shielders = length(unique\_ids)

```

#Unique shielding list

```{r}

#concatenate all individuals' group membership

shielding\_groups <-

 shielding\_long %>%

 group\_by(id) %>%

 arrange(group) %>%

 summarise(

 shielding\_groups = paste(group, collapse = ", "),

 shielding\_reasons = paste(shielding\_group\_description, collapse = ", "))

#add variable of source - if added at national record scan for any reason then national, else local

shielding\_source <-

 shielding\_long %>%

 group\_by(id) %>%

 summarise(

 central = sum(origin\_for\_this\_group == "central"),

 local = sum(origin\_for\_this\_group == "local")) %>%

 mutate(source = ifelse(central > 0, "National", "Local")) %>%

 select(id, source)

#unique patient characteristics

shielding\_people <-

 shielding\_long %>%

 group\_by(id) %>%

 slice\_head() %>%

 select(

 id,

 date\_addition,

 date\_removal,

 date\_death

 ) %>%

 left\_join(., shielding\_source, by = "id") %>%

 left\_join(., shielding\_groups, by = "id")

#add single shielding category for people who have multiple

#use hierarchy of cancer, transplant, respiratory, rare, immune, other (includes pregnant)

shielding\_people <-

 shielding\_people %>%

 mutate(

 dominant\_reason =

 case\_when(

 str\_detect(shielding\_reasons, "Cancer") ~ "Cancer",

 str\_detect(shielding\_reasons, "Transplant") ~ "Transplant",

 str\_detect(shielding\_reasons, "Respiratory") ~ "Respiratory",

 str\_detect(shielding\_reasons, "Rare") ~ "Rare disease",

 str\_detect(shielding\_reasons, "Immunosuppressants") ~ "Immunosuppressants",

 TRUE ~ "Other"

 )

 )

rm(shielding\_original, shielding\_groups, shielding\_source)

```

# Demographic data

```{r}

#load demographic data

demog <-

 vroom(here("original\_data", 'Dash416\_Shield20201012\_Demographic\_Release.csv'),

 delim = "¬") %>%

 rename(id = Dash416\_Release\_UID) %>%

 clean\_names() %>%

 select(id, calc\_sex, calc\_dob)

#load small area statistics about home

vulnerability <-

 vroom(here("original\_data", 'Dash416\_Shield20201012\_vulnerability\_Release.csv'),

 delim = "¬") %>%

 clean\_names() %>%

 rename(id = dash416\_release\_uid,

 ur\_name = u\_rname,

 ur\_class = u\_rclass,

 simd\_decile = simd2020v2\_decile,

 simd\_vigntile = simd2020v2\_vigintile,

 simd\_rank = simd2020v2\_rank) %>%

 #mutate(simd\_vigntile = as\_factor(simd\_vigntile)) %>%

 select(id, simd\_decile, simd\_vigntile, simd\_rank, ur\_class, ur\_name)

#keep only people in cohort

demog <-

 demog %>%

 filter(id %in% unique\_ids)

#create variables for sex and age band

#dob is only given as 1st of month of year born (disclosure control)

#age is calculated as of 1st March 2020

demog <- demog %>%

 mutate(

 sex = if\_else(calc\_sex == 1, "F", "M"),

 dob = ymd(calc\_dob),

 age = as.period(interval(dob, ymd("2020-03-01")), units = "years")$year,

 age\_band =

 case\_when(

 age %in% c(0:19) ~ "0-19",

 age %in% c(20:29) ~ "20-29",

 age %in% c(30:39) ~ "30-39",

 age %in% c(40:49) ~ "40-49",

 age %in% c(50:59) ~ "50-59",

 age %in% c(60:69) ~ "60-69",

 age %in% c(70:79) ~ "70-79",

 age %in% c(80:89) ~ "80-89",

 age >=90 ~ "90+")) %>%

 select(-calc\_sex,-calc\_dob)

#set age band as ordered factor

demog$age\_band <- factor(demog$age\_band, levels = c("0-19", "20-29", "30-39", "40-49", "50-59", "60-69", "70-79", "80-89", "90+"))

#add details of home area to demographics

demog <-

 demog %>%

 left\_join(., vulnerability, by = "id")

#add deaths to demog

demog <-

 shielding\_people %>%

 left\_join(demog, ., by = "id") %>%

 mutate(dead = ifelse(is.na(date\_death), 0, 1)) %>%

 relocate(date\_death, .after = last\_col())

#add some demographics values to long shielding list

shielding\_long <-

 demog %>%

 select(id, age, age\_band, sex, simd\_decile, ur\_name) %>%

 left\_join(shielding\_long, ., by = "id")

rm(vulnerability, shielding\_people)

#end result is two files: one (demog) with one case per person, and one (shielding\_long) with one case per reason shielding

```

# A&E data

```{r}

ae <-

 vroom(here(

 "original\_data",

 'Dash416\_HI\_ED\_Attendances\_Release\_v2\_1.csv'

 ),

 delim = "¬") %>%

 clean\_names() %>%

 rename(id = dash416\_release\_uid,

 date = arrival\_date)

#filter for cohort

ae <-

 filter(ae,

 id %in% unique\_ids)

#add demog data to A&E

ae <-

 ae %>%

 left\_join(., demog, by = "id")

```

#A&E 1 yr

```{r}

demog <-

ae %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 group\_by(id) %>%

 summarise(n\_ae\_1yr = n(),

 n\_ae\_major\_1yr = sum(major\_minor == "Major"),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

```

# Outpatient data

```{r}

#load and tidy Trak outpatient data

op <-

 vroom(here("original\_data", 'Dash416\_HI\_OP\_Admissions\_Release\_v2\_1.csv'), delim = "¬") %>%

 clean\_names() %>%

 rename(id = dash416\_release\_uid,

 date = appointment\_date,

 virtual = virtual\_non\_virtual\_appts,

 specialty = specialty\_local\_desc,

 specialty\_national = specialty\_national\_desc)

#exclusions for cohort and attended

op <-

 op %>%

 filter(id %in% unique\_ids) %>%

 filter(attendance\_status\_desc != "DNA") %>%

 select(-attendance\_status\_desc)

#add demog data to op

op <-

 op %>%

 left\_join(., demog, by = "id")

```

# Out-patient 1 & 5 year

```{r}

demog <-

 op %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 group\_by(id) %>%

 summarise(n\_op\_1yr = n(),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

demog <-

 op %>%

 filter(date >= ymd("2015-03-01"),

 date < ymd("2020-03-01")) %>%

 group\_by(id) %>%

 summarise(different\_clinics\_5yr = length(unique(specialty\_national)),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

```

# SMR inpatient data

```{r}

#load data, rename variables, format dates, add episode ID and date month, select only variables needed

smr01\_long\_original <-

 vroom(here("original\_data", 'Dash416\_Shield20201012\_SMR01\_Release.csv'),

 delim = "¬") %>%

 clean\_names() %>%

 rename(id = dash416\_release\_uid,

 main\_condition = main\_cod,

 other\_condition1 = oc1,

 other\_condition2 = oc2,

 other\_condition3 = oc3,

 other\_condition4 = oc4,

 other\_condition5 = oc5) %>%

 mutate(

 date\_episode\_start = as\_date(ymd(date\_episode\_start)),

 date\_episode\_end = as\_date(ymd(date\_episode\_end)),

 date\_admission = as\_date(ymd(adm\_date)),

 date\_discharge = as\_date(ymd(date\_discharge)),

 episode\_id = paste(id, date\_episode\_start, sep = "\_"))

#select date range and those in population

smr01\_long <-

 filter(smr01\_long\_original,

 date\_episode\_start >= ymd("2018-01-01"),

 id %in% unique\_ids)

```

# SMR01 single episode

```{r}

smr01\_flags<-

 smr01\_long %>%

 group\_by(episode\_id) %>%

 summarise(emergency = sum(emerg),

 itu = sum(itu),

 shdu = sum(shdu),

 ccu = sum(ccu),

 from\_care = sum(loc\_admission),

 to\_care = sum(loc\_discharge),

 .groups = "drop")

smr01\_los <-

 smr01\_long %>%

 group\_by(episode\_id) %>%

 slice\_head() %>%

 ungroup() %>%

 select(id, episode\_id, date\_episode\_start, date\_episode\_end) %>%

 mutate(episode\_los =

 difftime(date\_episode\_end, date\_episode\_start, units = "days"))

smr01 <-

 left\_join(smr01\_los, smr01\_flags, by = "episode\_id")

#add demog data to hospital admissions

smr01 <-

 smr01 %>%

 mutate(date = date\_episode\_start) %>%

 left\_join(., demog, by = "id")

rm(smr01\_flags, smr01\_los)

```

# SMR01 1 yr

```{r}

demog <-

smr01 %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 group\_by(id) %>%

 summarise(n\_smr01\_1yr = n(),

 n\_emerg\_1yr = sum(emergency == 1),

 n\_routine\_1yr = n\_smr01\_1yr - n\_emerg\_1yr,

 los\_smr01\_1yr = sum(episode\_los),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

```

# Tidy 1 year care

```{r}

#change na to 0

#sum op + routine ip as "routine" and ae + emergency ip as "emergency"

demog <-

 demog %>%

 mutate(

 across(n\_ae\_1yr:los\_smr01\_1yr, ~ replace\_na(., 0)),

 all\_routine\_1yr = n\_op\_1yr + n\_routine\_1yr,

 all\_emergency\_1yr = n\_ae\_1yr + n\_emerg\_1yr,

 prop\_care\_emergency\_1yr =

 round(all\_emergency\_1yr / (all\_emergency\_1yr + all\_routine\_1yr), 2))

```

# Morbidity

```{r}

#make long df that has one diagnosis per row (comorbidity input)

#for 5 years before lockdown

smr01\_diagnoses <-

 smr01\_long\_original %>%

 filter(date\_episode\_start >= ymd("2015-03-01"),

 date\_episode\_start < ymd("2020-03-01"),

 id %in% unique\_ids) %>%

 pivot\_longer(main\_condition:other\_condition5, values\_drop\_na = T) %>%

 select(id, episode\_id, date\_episode\_start, name, value)

#use comorbidity package to calculate chronic diseases

person\_morbidity <-

 comorbidity(

 x = smr01\_diagnoses,

 id = "id",

 code = "value",

 score = "elixhauser",

 icd = "icd10",

 assign0 = T

 ) %>%

 mutate(score\_band = case\_when(score == 0 ~ 0,

 score == 1 ~ 1,

 score > 1 ~ 2))

#add chronic disease data to demographics

demog <-

demog %>%

 left\_join(., person\_morbidity, by = "id")

rm(person\_morbidity, smr01\_long\_original, smr01\_diagnoses)

```

#Grampian data

```{r}

#in-patient

gramp\_ip <-

 vroom(here("original\_data", 'vulnerable\_416\_All\_IP\_Admsssions\_Aggregate.csv'), delim = "¬") %>%

 select(-Visit\_Status\_Desc) %>%

 rename(date = AdmDate,

 type = Adm\_Type,

 total = no\_adms) %>%

 mutate(type = ifelse(type == "Non-Elective", "emergency", "scheduled"))

#out-patient

gramp\_op <- vroom(here("original\_data", 'vulnerable\_416\_All\_OP\_Attendances\_Aggregate.csv'), delim = "¬") %>%

 select(-Attendance\_Status\_Desc) %>%

 rename(date = Appointment\_Date,

 virtual = "Virtual/Non Virtual appts",

 total = no\_appts) %>%

 mutate(virtual = ifelse(virtual == 1,"virtual", "in person"))

#emergency

gramp\_ae <- vroom(here("original\_data", 'vulnerable\_416\_All\_ED\_Admissions\_Aggregate.csv'), delim = ",") %>%

 select(-Visit\_Type\_Desc) %>%

 select(Arrival\_Date, everything()) %>%

 rename(date = Arrival\_Date,

 department = Emergency\_Dept\_Desc,

 arrival = Arrival\_Mode,

 admitted = "Hospital admission",

 total = ED\_Atttend)

```

#Time periods

```{r}

#function to fix dates and calculate time periods

fix\_dates <-

 function(input\_file) {

 mutate(input\_file,

 date = as\_date(date),

 date\_week = floor\_date(date, unit = "week"),

 date\_month = floor\_date(date, unit = "month"),

 date\_season = floor\_date(date, unit = "season"),

 date\_year = floor\_date(date, unit = "year"))

 }

op <- fix\_dates(op)

smr01 <- fix\_dates(smr01)

ae <- fix\_dates(ae)

gramp\_op <- fix\_dates(gramp\_op)

gramp\_ip <- fix\_dates(gramp\_ip)

gramp\_ae <- fix\_dates(gramp\_ae)

```

#Grampian summary

```{r}

gramp\_op %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 summarise(n\_gramp\_op\_1yr = sum(total))

gramp\_ip %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 group\_by(type) %>%

 summarise(n\_gramp\_ip\_1yr = sum(total), .groups = "drop")

gramp\_ae %>%

 filter(date >= ymd("2019-03-01"),

 date < ymd("2020-03-01")) %>%

 summarise(n\_gramp\_ae\_1yr = sum(total))

```

#Usage summaries

```{r}

smr01\_admissions <-

 smr01 %>%

 group\_by(date\_month) %>%

 summarise(total = n(),

 emergency = sum(emergency),

 routine = total - emergency,

 .groups = "drop") %>%

 mutate(across(total:routine,

 .fns = ~./n\_shielders \* 100,

 .names = "{col}\_per\_100")) %>%

 mutate(across(contains("per\_100"), round, 1))

op\_attendances <-

 op %>%

 group\_by(date\_month) %>%

 summarise(n\_attendances = n(),

 n\_virtual = sum(virtual),

 n\_new = sum(referral\_type == "New"),

 n\_return = sum(referral\_type == "Return"),

 n\_adhoc = sum(referral\_type == "Adhoc"),

 .groups = "drop")

specialties\_2019 <-

 op %>%

 filter(date\_year == "2019-01-01") %>%

 filter(specialty\_national != "Electrocardiography") %>%

 count(specialty\_national, sort = T) %>%

 slice(1:15)

top\_15 <- specialties\_2019 %>% slice(1:15) %>% .$specialty\_national

op\_clinics <-

 op %>%

 filter(specialty\_national %in% top\_15) %>%

 group\_by(date\_month, specialty\_national) %>%

 summarise(n = n(),

 .groups = "drop")

ae\_admissions <-

 ae %>%

 group\_by(date\_month) %>%

 summarise(total = n(),

 major = sum(major\_minor == "Major"),

 minor = sum(major\_minor == "Minor"),

 .groups = "drop")

```

#Lockdown usage per person

```{r}

demog <-

 smr01 %>%

 filter(date >= as\_date("2020-04-01")) %>%

 filter(date < as\_date("2020-08-01")) %>%

 group\_by(id) %>%

 summarise(n\_lockdown\_admissions = n(),

 n\_lockdown\_admissions\_emerg = sum(emergency),

 n\_lockdown\_admissions\_routine = sum(emergency == 0),

 n\_lockdown\_admissions\_itu = sum(itu),

 n\_lockdown\_admissions\_shdu = sum(shdu),

 n\_lockdown\_admissions\_ccu = sum(ccu),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

demog <-

 op %>%

 filter(date >= as\_date("2020-04-01")) %>%

 filter(date < as\_date("2020-08-01")) %>%

 group\_by(id) %>%

 summarise(

 n\_lockdown\_op = n(),

 n\_lockdown\_op\_virtual = sum(virtual),

 n\_lockdown\_op\_new = sum(referral\_type == "New", na.rm = T),

 n\_lockdown\_op\_return = sum(referral\_type == "Return", na.rm = T),

 n\_lockdown\_op\_adhoc = sum(referral\_type == "Adhoc", na.rm = T),

 n\_lockdown\_op\_specialties = n\_distinct(specialty),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

demog <-

 ae %>%

 filter(date >= as\_date("2020-04-01")) %>%

 filter(date < as\_date("2020-08-01")) %>%

 group\_by(id) %>%

 summarise(n\_lockdown\_ae = n(),

 n\_lockdown\_ae\_major = sum(major\_minor == "Major"),

 n\_lockdown\_ae\_hosp = sum(str\_detect(departure\_destination\_desc, "NHS")),

 .groups = "drop") %>%

 left\_join(demog, ., by = "id")

#is this right?

demog <-

 demog %>%

 mutate(

 across(contains("lockdown"), ~ replace\_na(., 0)))

```

#Figures

#Source figs

```{r}

source\_fig <-

 demog %>%

 mutate(dominant\_reason = fct\_rev(fct\_infreq(dominant\_reason))) %>%

 rename(Identified = source) %>%

 mutate(Identified = ifelse(Identified == "National", "Nationally", "Grampian")) %>%

 ggplot(aes(x = dominant\_reason, fill = Identified)) + geom\_bar() +

 scale\_fill\_manual(values = c("black", "gray30")) +

 coord\_flip() + labs(x = "", y = "") + theme(legend.position = c(0.7, 0.2), axis.text = element\_text(size = 14))

simd\_source <-

demog %>%

 filter(!is.na(simd\_rank)) %>%

 rename(Identified = source) %>%

 mutate(Identified = ifelse(Identified == "National", "Nationally", "Locally")) %>%

 ggplot(aes(x = simd\_rank, fill = Identified)) +

 geom\_histogram(bins = 50, position = "fill") +

 scale\_fill\_manual(values = c("dark gray", "black")) +

 labs(y = "Home Area Deprivation Rank", x = "") +

 scale\_x\_reverse(

 breaks = c(275, 2500, 5000, 7000),

 labels = c("1\nmost\ndeprived", "2500", "5000", "7000\nleast\n deprived")

 ) + labs(y = "Proportion identified", x = "Home Area Deprivation Rank") + theme(legend.position = c(0.75, 0.2), legend.key = element\_rect(fill = "white"), legend.text = element\_text(color = "white"), legend.title = element\_text(color = "white"))

age\_source\_fig <-

demog %>%

 filter(age <= 99) %>%

 rename(Identified = source) %>%

 mutate(Identified = ifelse(Identified == "National", "Nationally", "Locally")) %>%

 ggplot(aes(x = age, fill = Identified)) +

 geom\_histogram(binwidth = 2, position = "fill") +

 labs(y = "Proportion identified", x = "Age") +

 scale\_fill\_manual(values = c("dark gray", "black")) +

 theme(legend.position = c(0.75, 0.2), legend.key = element\_rect(fill = "white"), legend.text = element\_text(color = "white"), legend.title = element\_text(color = "white"))

```

```{r, fig.height=8}

plot\_grid(source\_fig, age\_source\_fig, simd\_source,

 labels = "AUTO")

```

#Demog figs

```{r}

sex\_fig <-

demog %>%

 group\_by(dominant\_reason, sex) %>%

 summarise(n = n(),

 .groups = "drop") %>%

 mutate(dominant\_reason = fct\_reorder(dominant\_reason, n)) %>%

 ggplot(aes(x = dominant\_reason, y = n, fill = sex)) +

 geom\_bar(stat = "identity", position= "dodge")+

 coord\_flip() +

 labs(x = "", y = "People") +

 theme(legend.position = c(0.8, 0.2), legend.title = element\_blank()) +

 scale\_fill\_manual(values = c("gray40", "gray")) +

 guides(fill = guide\_legend(reverse = T))

age\_fig <-

demog %>%

 mutate(dominant\_reason = fct\_reorder(dominant\_reason, age)) %>%

 ggplot(aes(y = age, x = dominant\_reason, fill = sex)) +

 geom\_boxplot(width = 0.5, size = 1, outlier.shape = NA, coef = 0) +

 coord\_flip(ylim = c(0, 90)) +

 labs(x = "", y = "Age") +

 theme(legend.position = c(0.25, 0.9), legend.title = element\_blank()) +

 scale\_fill\_manual(values = c("gray", "white")) +

 guides(fill = guide\_legend(reverse = T))

simd\_fig <-

demog %>%

 filter(!is.na(simd\_decile)) %>%

 ggplot(aes(simd\_decile, fill = dominant\_reason)) +

 geom\_bar(position = "fill") +

 scale\_x\_reverse(

 breaks = c(1, 10),

 labels = c("Most\ndeprived", "Least\ndeprived")) +

 theme(legend.title = element\_blank()) +

 labs(y = "Proportion", x = "Home Area Deprivation Deciles")

```

```{r, fig.height=8, fig.width=10}

top\_row <- plot\_grid(sex\_fig, age\_fig, labels = c("A", "B"))

plot\_grid(top\_row, simd\_fig, ncol = 1, labels = c("", "C"))

```

#Emerg Routine fig

```{r}

visit\_type\_reason\_fig <-

demog %>%

 select(id, dominant\_reason, all\_emergency\_1yr, all\_routine\_1yr) %>%

 rename(Emergency = all\_emergency\_1yr, Scheduled = all\_routine\_1yr) %>%

 pivot\_longer(-c(id, dominant\_reason), names\_to = "care\_type", values\_to = "visits") %>%

 group\_by(dominant\_reason, care\_type) %>%

 summarise(n = sum(visits),

 n\_people = n(),

 mean = mean(visits),

 .groups = "drop") %>%

 mutate(dominant\_reason = fct\_reorder(dominant\_reason, mean),

 care\_type = factor(care\_type, levels = c("Scheduled", "Emergency"))) %>%

 ggplot() +

 #geom\_point(aes(x = dominant\_reason, y = mean), size = 2) +

 geom\_bar(aes(x = dominant\_reason, y = mean), stat = "identity") +

 labs(x = "", y = "Pre-pandemic\n annual visits\nper person\n") +

 facet\_wrap(~care\_type, scales = "free") +

 theme(legend.position = "none", strip.background = element\_rect(fill = "white")) +

 theme(axis.text.x = element\_text(angle = 35, hjust = 1)) +

 expand\_limits(y = 0)

visit\_type\_age\_fig <-

demog %>%

 select(id, age\_band, all\_emergency\_1yr, all\_routine\_1yr) %>%

 rename(Emergency = all\_emergency\_1yr, Scheduled = all\_routine\_1yr) %>%

 pivot\_longer(-c(id, age\_band), names\_to = "care\_type", values\_to = "visits") %>%

 group\_by(age\_band, care\_type) %>%

 summarise(n = sum(visits),

 n\_people = n(),

 mean = mean(visits),

 .groups = "drop") %>%

 mutate(care\_type = factor(care\_type, levels = c("Scheduled", "Emergency"))) %>%

 ggplot() +

 geom\_bar(aes(x = age\_band, y = mean), stat = "identity") +

 labs(x = "", y = "Pre-pandemic\n annual visits\nper person\n") +

 facet\_wrap(~care\_type, scales = "free") +

 theme(legend.position = "none", strip.background = element\_rect(fill = "white")) +

 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +

 expand\_limits(y = 0)

```

```{r, fig.height=8, fig.width=8}

plot\_grid(visit\_type\_reason\_fig, visit\_type\_age\_fig, ncol = 1, labels = "AUTO", rel\_heights = c(1.2, 1))

```

#SMR01 fig

```{r}

#smr01\_fig <-

 smr01\_admissions %>%

 filter(date\_month <= ymd("2020-07-01")) %>%

 pivot\_longer(cols = -date\_month,

 names\_to = "admission\_type",

 values\_to = "admissions") %>%

 filter(admission\_type %in% c("emergency", "routine")) %>%

 ggplot(aes(x = date\_month, y = admissions, color = admission\_type)) +

 geom\_vline(xintercept = as\_date("2020-03-26"), color = "grey") +

 geom\_vline(xintercept = as\_date("2020-06-18"), color = "grey") +

 geom\_line(aes(linetype = admission\_type), size = 1, color = "black") +

 ylim(0, 100) +

 labs(x = "", y = "In-patient admissions / month") +

 scale\_linetype\_manual(values = c("solid", "twodash")) +

 theme(legend.title = element\_blank(),

 legend.position = c(0.2, 0.2))

```

# OP fig

```{r}

#op\_virtual\_fig <-

 op\_attendances %>%

 filter(date\_month >= ymd("2019-01-01")) %>%

 filter(date\_month < ymd("2021-01-01")) %>%

 mutate(n\_virtual = ifelse(n\_virtual < 5, NA\_real\_, n\_virtual)) %>%

 pivot\_longer(cols = -date\_month,

 names\_to = "attendance\_type",

 values\_to = "attendances") %>%

 filter(attendance\_type %in% c("n\_attendances", "n\_virtual")) %>%

 filter(attendances >= 5) %>%

 ggplot(aes(x = date\_month, y = attendances, color = attendance\_type)) +

 geom\_line(size = 1) +

 scale\_x\_date(date\_breaks = "5 month", date\_labels = "%b %Y") +

 labs(x = "", y = "Out-patient\n attendances per month\n") +

 geom\_vline(xintercept = as\_date("2020-03-26"), color = "grey") +

 geom\_vline(xintercept = as\_date("2020-07-21"), color = "grey") +

 scale\_color\_manual(values = c("black", "dark orange"), labels = c("Total Outpatient", "Virtual Visit")) +

 theme(legend.title = element\_blank(), legend.position = "bottom", axis.text.x = element\_text(size = 10))

```

#Clinics fig

```{r, fig.height=6, fig.width=7}

#op\_clinics\_fig <-

 op\_clinics %>%

 filter(date\_month < ymd("2020-08-01")) %>%

 filter(date\_month >= ymd("2019-01-01")) %>%

 ggplot(aes(x = date\_month, y = n)) +

 geom\_line() +

 geom\_vline(xintercept = as\_date("2020-03-26"), color = "grey") +

 geom\_vline(xintercept = as\_date("2020-07-31"), color = "grey") +

 labs(x = "", y = "") +

 facet\_wrap(

 ~ fct\_rev(fct\_reorder(specialty\_national, n, max)),

 ncol = 3,

 scales = "free",

 labeller = label\_wrap\_gen()) +

 expand\_limits(y = 0) +

 theme(legend.title = element\_blank(),

 axis.text = element\_text(size = 8),

 strip.text = element\_text(size = 10),

 strip.background = element\_rect(fill = "white"))

```

#AE fig

```{r}

ae\_admissions %>%

 #filter(date\_month <= ymd("2020-07-01")) %>%

 ggplot(aes(x = date\_month, y = total)) +

 geom\_vline(xintercept = as\_date("2020-03-26"), color = "grey") +

 geom\_vline(xintercept = as\_date("2020-06-18"), color = "grey") +

 geom\_line(size = 1, color = "black") +

 ylim(0, 1000) +

 labs(x = "", y = "A&E attendances / month")

```

#Tables

#Demog Table Function

```{r}

#create summary table from dataframe

create\_summary\_table <-

 function(x) {

 x %>%

 summarise(

 n\_people = n(),

 prop\_shielding = n\_people/n\_shielders,

 n\_women = sum(sex == "F"),

 prop\_women = n\_women / n\_people,

 median\_age = median(age),

 under\_20 = sum(age < 20),

 prop\_children = under\_20 / n\_people,

 median\_simd = median(simd\_decile, na.rm = T),

 simd\_1\_or\_2 = sum(simd\_decile == 1 | simd\_decile == 2, na.rm = T),

 prop\_deprived = simd\_1\_or\_2 / n\_people,

 remote\_or\_rural = sum(ur\_class %in% c(4:6)),

 prop\_remote\_rural = remote\_or\_rural / n\_people,

 n\_national = sum(source == "National"),

 prop\_national = n\_national/n\_people,

 prop\_multimorbid = sum(score > 1, na.rm = T) / n\_people,

 mean\_wscore\_vw = mean(wscore\_vw, na.rm = T),

 .groups = "drop") %>%

 mutate(across(starts\_with("prop"), round, 2))

 }

```

#Demographics tables

```{r}

demog %>%

 create\_summary\_table() %>%

 write\_csv("demog\_totals.csv")

demog %>%

 group\_by(age\_band) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_age.csv")

demog %>%

 group\_by(sex) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_sex.csv")

demog %>%

 group\_by(dominant\_reason) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_reason.csv")

demog %>%

 group\_by(source) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_source.csv")

demog %>%

 group\_by(simd\_decile) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_simd.csv")

demog %>%

 group\_by(ur\_name) %>%

 create\_summary\_table() %>%

 write\_csv("demog\_ur.csv")

```

#Precovid Care Summary Function

```{r}

#create summary table from dataframe

create\_precovid\_table <-

 function(x) {

 x %>%

 summarise(

 n\_people = n(),

 prop\_shielding = n\_people/n\_shielders,

 prop\_op = sum(n\_op\_1yr > 0) / n\_people,

 prop\_ip\_scheduled = sum(n\_routine\_1yr > 0) / n\_people,

 prop\_ip\_emergency = sum(n\_emerg\_1yr > 0) / n\_people,

 prop\_ae = sum(n\_ae\_1yr > 0) / n\_people,

 n\_all\_routine = sum(all\_routine\_1yr),

 mean\_routine = mean(all\_routine\_1yr),

 n\_all\_emergency = sum(all\_emergency\_1yr),

 mean\_emergency = mean(all\_emergency\_1yr),

 prop\_emergency = n\_all\_emergency / (n\_all\_emergency + n\_all\_routine),

 los\_year = sum(los\_smr01\_1yr),

 mean\_los = mean(los\_smr01\_1yr),

 prop\_multimorbid = sum(score > 1, na.rm = T) / n\_people,

 mean\_wscore\_vw = mean(wscore\_vw, na.rm = T),

 .groups = "drop") %>%

 mutate(across(starts\_with("prop"), round, 2)) %>%

 mutate(across(contains("mean"), round, 1))

 }

```

#Pre-Covid care tables

```{r}

demog %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_totals.csv")

demog %>%

 group\_by(age\_band) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_age.csv")

demog %>%

 group\_by(sex) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_sex.csv")

demog %>%

 group\_by(dominant\_reason) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_reason.csv")

demog %>%

 group\_by(source) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_source.csv")

demog %>%

 group\_by(simd\_decile) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_simd.csv")

demog %>%

 group\_by(ur\_name) %>%

 create\_precovid\_table() %>%

 write\_csv("precovid\_ur.csv")

```

#Lockdown summaries

```{r}

create\_lockdown\_usage\_table <-

 function(x) {

 x %>%

 summarise(

 n\_people = n(),

 prop\_op = sum(n\_lockdown\_op > 0) / n\_people,

 prop\_smr01\_routine = sum(n\_lockdown\_admissions\_routine > 0) / n\_people,

 prop\_smr01\_emerg = sum(n\_lockdown\_admissions\_emerg > 0) / n\_people,

 prop\_ae = sum(n\_lockdown\_ae > 0) / n\_people,

 mean\_op = mean(n\_lockdown\_op) \* 100,

 mean\_smr01\_routine = mean(n\_lockdown\_admissions\_routine) \* 100,

 mean\_smr01\_emerg = mean(n\_lockdown\_admissions\_emerg) \* 100,

 mean\_ae = mean(n\_lockdown\_ae) \* 100,

 n\_op = sum(n\_lockdown\_op),

 n\_smr01\_routine = sum(n\_lockdown\_admissions\_routine),

 n\_smr01\_emerg = sum(n\_lockdown\_admissions\_emerg),

 n\_ae = sum(n\_lockdown\_ae),

 n\_total\_routine = n\_op + n\_smr01\_routine,

 n\_total\_emergency = n\_ae + n\_smr01\_emerg,

 prop\_emergency = n\_total\_emergency / (n\_total\_routine + n\_total\_emergency),

 n\_dead = sum(dead == 1),

 dead\_per\_k = round(n\_dead / n\_people \* 1000, 0),

 .groups = "drop") %>%

 mutate(across(contains("mean"), round, 0)) %>%

 mutate(across(contains("prop"), round, 2))

 }

```

#Who lockdown care

```{r}

demog %>%

 group\_by(n\_lockdown\_ae > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_ae.csv")

demog %>%

 group\_by(n\_lockdown\_ae\_major > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_ae\_major.csv")

demog %>%

 group\_by(n\_lockdown\_admissions > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_smr01.csv")

demog %>%

 group\_by(n\_lockdown\_admissions\_emerg > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_smr01\_emerg.csv")

demog %>%

 group\_by(n\_lockdown\_admissions\_routine > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_smr01\_routine.csv")

demog %>%

 group\_by(n\_lockdown\_op > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_op.csv")

demog %>%

 group\_by(n\_lockdown\_op\_virtual > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_op\_virtual.csv")

demog %>%

 group\_by(n\_lockdown\_op\_return > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_op\_return.csv")

demog %>%

 group\_by(n\_lockdown\_op\_new > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_op\_new.csv")

demog %>%

 group\_by(n\_lockdown\_op\_adhoc > 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_op\_adhoc.csv")

demog %>%

 group\_by(no\_visits = n\_lockdown\_admissions == 0 & n\_lockdown\_op == 0 & n\_lockdown\_ae == 0) %>%

 create\_summary\_table() %>%

 write\_csv(., "lockdown\_usage\_no\_visits.csv")

```

#What lockdown care

```{r}

demog %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_total.csv")

demog %>%

 group\_by(source) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_source.csv")

demog %>%

 group\_by(sex) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_sex.csv")

demog %>%

 group\_by(age\_band) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_age.csv")

demog %>%

 group\_by(simd\_decile) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_simd.csv")

demog %>%

 group\_by(dominant\_reason) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_reason.csv")

demog %>%

 group\_by(ur\_name) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_ur.csv")

demog %>%

 group\_by(index) %>%

 create\_lockdown\_usage\_table() %>%

 write\_csv(., "lockdown\_usage\_mm.csv")

```

# Health Care Resource Utilisation (HCRU) by Grampian vs Shielding populations

# Generalised Additive Model using mgcv library

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# library ----

library(data.table)

library(bit64)

library(mgcv)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Data ----

# Data start date, lockdown start date and lockdown end date

dsDate <- as.IDate('2020-01-12')

lsDate <- as.IDate('2020-03-15')

leDate <- as.IDate('2020-06-18')

exclDate <- as.IDate('2020-03-15')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Fn to organise data -----

fn\_createData <- function(oDT){

 # Subset data

 wDT <- oDT[date\_week <= leDate,]

 # Reshape data

 setcolorder(wDT, c('date\_week',

 'grampian\_total', 'shielding\_total',

 'grampian\_total\_per\_k', 'shielding\_total\_per\_k'))

 value\_name <- c('total', 'total\_per\_k')

 DT <- melt(wDT, id.vars = 'date\_week',

 measure.vars = patterns('.\*total$', 'total\_per\_k'),

 variable.name = 'Group',

 value.name = value\_name)

 # Group variable

 DT$Group <- as.character(DT$Group)

 DT[, Group := ifelse(Group == '1', 'G', 'S')]

 # Create Phase variable

 DT$Phase <- NA\_character\_

 DT[, Phase := ifelse(date\_week <= lsDate, 'P1', 'P2')]

 # str(DT)

 # Group\_Phase

 DT[, Group\_Phase := as.factor(paste0(Group, '\_', Phase))]

 # Week

 DT[, iWeek := as.integer((date\_week - dsDate) %/% 7 + 1)]

 # Integer N

 DT[, N := as.integer(total)]

 DT[Group == 'G', NP := 585700]

 DT[Group == 'S', NP := 16092]

 # Remove Week 10

 DT <- DT[!(iWeek == 10), ]

 return(DT)

}

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Outpatient ----

# ALL OPD (TRAK)

fname <- 'trak\_outpatient\_per\_week\_total.csv'

OPD <- fread(file = paste0(datDir, fname), sep = ',')

OPD$date\_week <- as.IDate(OPD$date\_week)

OPD <- OPD[date\_week != exclDate & date\_week >= dsDate & date\_week <= leDate,]

DT <- fn\_createData(OPD)

# FINAL MODEL

fm\_NB <- mgcv::gam(N ~ Group\_Phase + iWeek + offset(log(NP)),

 family = nb(link = 'log'),

 method = 'REML', gamma = 1,

 na.action = na.omit,

 data = DT[!(iWeek == 10), ])

summary(fm\_NB)

anova(fm\_NB)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Inpatient Non-Emergency (Scheduled) ----

# Data

fname <- 'smr01\_inpatient\_per\_week\_total.csv'

IP <- fread(file = paste0(datDir, fname), sep = ',')

IP$date\_week <- as.IDate(IP$date\_week)

IP <- IP[date\_week != exclDate & date\_week >= dsDate & date\_week <= leDate,]

IP <- IP[admission\_type == 'scheduled', ]

DT <- fn\_createData(IP)

# FINAL MODEL

fm\_NB <- mgcv::gam(N ~ Group\_Phase + iWeek + offset(log(NP)),

 family = nb(link = 'log'),

 method = 'REML', gamma = 1,

 na.action = na.omit,

 data = DT[!(iWeek == 10), ])

summary(fm\_NB)

anova(fm\_NB)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Inpatient Emergency ----

# Data

fname <- 'smr01\_inpatient\_per\_week\_total.csv'

IP <- fread(file = paste0(datDir, fname), sep = ',')

IP$date\_week <- as.IDate(IP$date\_week)

IP <- IP[date\_week != exclDate & date\_week >= dsDate & date\_week <= leDate,]

IP <- IP[admission\_type == 'emergency', ]

DT <- fn\_createData(IP)

# FINAL MODEL

fm\_NB <- mgcv::gam(N ~ Group\_Phase + iWeek + offset(log(NP)),

 family = nb(link = 'log'),

 method = 'REML', gamma = 1,

 na.action = na.omit,

 data = DT[!(iWeek == 10), ])

summary(fm\_NB)

anova(fm\_NB)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Accident & Emergency -----

# ALL AE (TRAK)

fname <- 'trak\_emergency\_per\_week\_total.csv'

AE <- fread(file = paste0(datDir, fname), sep = ',')

AE$date\_week <- as.IDate(AE$date\_week)

AE <- AE[date\_week != exclDate & date\_week >= dsDate & date\_week <= leDate,]

DT <- fn\_createData(AE)

# FINAL MODEL

fm\_NB <- mgcv::gam(N ~ Group\_Phase + iWeek,

 family = nb(link = 'log'),

 method = 'REML', gamma = 1,

 na.action = na.omit, nthreads = nt,

 data = DT[!(iWeek == 10), ])

summary(fm\_NB)

anova(fm\_NB)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Fitting GLMM on summarised data of Shielding patients

# Uses summarised data for the combination of: (Phase, sex, AGEgr\_, SIMDgr\_, SHgroup\_)

# Fit using glmmTMB library

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# library ----

library(data.table)

library(glmmTMB)

library(parallel)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# OPD ----

# OPD

load('S\_OPD.RData')

# Data summary

DT <- fDT[, .(nEvent = sum(N), nPat = .N, iDay = median(iDay)),

 by = .(Phase, sex, AGEgr\_, SIMDgr\_, SHgroup\_)]

DT$gID <- as.factor(paste0('R', rep(1:(nrow(DT)/2), each = 2)))

# Intermediate Model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 sex:Phase + SIMDgr\_:Phase + AGEgr\_:Phase + SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_OPD\_NB\_int.RData')

# Final Model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 sex:Phase + AGEgr\_:Phase + SHgroup\_:Phase +

 AGEgr\_:SHgroup\_ + AGEgr\_:SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_OPD\_NB\_final.RData')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# IP Non-Emergency (Scheduled) ----

# IP

load('S\_IP.RData')

fDT$N <- fDT$N\_NonEmerg

# Data summary

DT <- fDT[, .(nEvent = sum(N), nPat = .N, iDay = median(iDay)),

 by = .(Phase, sex, AGEgr\_, SIMDgr\_, SHgroup\_)]

DT$gID <- as.factor(paste0('R', rep(1:(nrow(DT)/2), each = 2)))

# Intermediate Model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 sex:Phase + SIMDgr\_:Phase + AGEgr\_:Phase + SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_IP\_NonEmerg\_NB\_int.RData')

# Final model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 AGEgr\_:Phase + SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_IP\_NonEmerg\_NB\_final.RData')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# IP Emergency----

# IP

load('S\_IP.RData')

fDT$N <- fDT$N\_Emerg

# Data summary

DT <- fDT[, .(nEvent = sum(N), nPat = .N, iDay = median(iDay)),

 by = .(Phase, sex, AGEgr\_, SIMDgr\_, SHgroup\_)]

DT$gID <- as.factor(paste0('R', rep(1:(nrow(DT)/2), each = 2)))

# Intermediate Model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 sex:Phase + SIMDgr\_:Phase + AGEgr\_:Phase + SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_IP\_Emerg\_NB\_int.RData')

# Final model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom1(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_IP\_Emerg\_NB\_final.RData')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Accident & Emergency ----

# AE

load('S\_AE.RData')

# Data summary

DT <- fDT[, .(nEvent = sum(N), nPat = .N, iDay = median(iDay)),

 by = .(Phase, sex, AGEgr\_, SIMDgr\_, SHgroup\_)]

DT$gID <- as.factor(paste0('R', rep(1:(nrow(DT)/2), each = 2)))

# Intermediate Model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 sex:Phase + SIMDgr\_:Phase + AGEgr\_:Phase + SHgroup\_:Phase +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom2(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_AE\_NB\_int.RData')

# Final model (NB)

fm\_NB <- glmmTMB::glmmTMB(nEvent ~ sex + SIMDgr\_ + AGEgr\_ + SHgroup\_ + Phase +

 AGEgr\_:Phase + AGEgr\_:SHgroup\_ +

 offset(log(iDay)) + offset(log(nPat)) + (1 | gID),

 family = nbinom1(link = 'log'),

 data = DT,

 control = glmmTMBControl(parallel = parallel::detectCores()))

summary(fm\_NB)

fm <- fm\_NB

save(fm, file = 'fm\_AE\_NB\_final.RData')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_