

# Vaccines and patient characteristics

## Import libraries and data

The datasets used for this report are created using the study definition [/analysis/study\\_definition.py](#) [\(..analysis/study\\_definition.py\)](#), using codelists referenced in [/codelists/codelists.txt](#) [\(..codelists/codelists.txt\)](#).

```
In [1]: %load_ext autoreload
        %autoreload 2

import pyodbc
import pandas as pd
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime, date
import os
import glob
import subprocess
from ebmdatalab import charts
from IPython.display import display, Markdown, HTML
#from lib.functions import *

# get current branch
current_branch = subprocess.run(["git", "rev-parse", "--abbrev-ref", "HEAD"], capture_output=True).stdout.decode("utf8").strip()
```

```

In [2]: # import data, carry out basic joins for lookups and create flags.

# import data and fill nulls with 0
df = pd.read_csv(os.path.join("../", "output", "input_delivery.csv")).fillna(0)

# ensure that no patients are included outside the desired criteria
# df = df.loc[(df["care_home_type"].isin(["PS", "PN", "PC"])) | (df["ageband"].isin(["70-79", "80+"]) |
# (df["covid_vacc_date"]!=0))]

# convert ethnic categories to words
eth_16 = pd.read_csv(os.path.join("../", "analysis", "ethnicity_16_lookup.csv"))
df = df.merge(eth_16, how="left", left_on="ethnicity_16", right_on="code").rename(columns={"name": "Eth
nicity_sixteen_categories"})

ethnicity_6_lookup = {0: "Unknown", 1: "White", 2: "Mixed", 3: "South Asian", 4: "Black", 5: "Other"}
df["Ethnicity_broad_categories"] = [ethnicity_6_lookup[x] for x in df['ethnicity']].fillna(0).astype(in
t)]

# describe imd partially in words
imd_lookup = {0: "Unknown", 1: "1 Most deprived", 2: "2", 3: "3", 4: "4", 5: "5 Least deprived"}

# replace flags with descriptive terms
df = df.assign(
    covid_vacc_flag = np.where(df["covid_vacc_date"]!=0, "vaccinated", "unvaccinated"),
    covid_vacc_flag_ox = np.where(df["covid_vacc_oxford_date"]!=0, 1, 0),
    covid_vacc_flag_pfz = np.where(df["covid_vacc_pfizer_date"]!=0, 1, 0),
    covid_vacc_2nd = np.where(df["covid_vacc_second_dose_date"]!=0, 1, 0),
    covid_vacc_bin = np.where(df["covid_vacc_date"]!=0, 1, 0),
    ##care_home = np.where(df["care_home_type"].isin(["PS", "PN", "PC"]), "care home", "other residence
s"),
    ssri = np.where((df["ssri"]==1) & (df["psychosis_schiz_bipolar"]==0) & \
        (df["intel_dis_incl_downs_syndrome"]==0) & (df["dementia"]==0), 1, 0),
    Ethnicity_broad_categories = df['Ethnicity_broad_categories'].fillna("Unknown"),
    Ethnicity_sixteen_categories = df['Ethnicity_sixteen_categories'].fillna("Unknown"),
    region = df['region'].replace(0, "Unknown"),
    stp = df['stp'].replace(0, "Unknown"),
    sex = df['sex'].replace(['I', 'U'], "Other/Unknown"),
    bmi = np.where((df["bmi"]=="Not obese"), "under 30", "30+"),
    Index_of_Multiple_Deprivation = [imd_lookup[x] for x in df['imd']].fillna(0).astype(int))
)\
    .drop(["imd", "code", "ethnicity_16", "ethnicity", "adrenaline_pen", "has_died", "has_follow_up"], 1)

# care homes: regroup age bands (to later keep only 65+ labelled as care home residents)
df.loc[(df["care_home_type"].isin(["PS", "PN", "PC"])) & (df["age"]>=65) & (df["age"]<70), "ageband"] =
"65-69"
# amend community age band to remove any care home flags for under 65s
# (only elderly care homes are included so these are likely live-in staff+their families or other n
on-care recipients)
df.loc[(df["ageband_community"]=="care home") & (df["age"]<65), "ageband_community"] = df["ageband"]

for c in ["care_home", "dementia",
    "chronic_cardiac_disease", "current_copd", "dialysis", "dmards", "psychosis_schiz_bipolar",
    "solid_organ_transplantation", "chemo_or_radio", "intel_dis_incl_downs_syndrome", "ssri",
    "lung_cancer", "cancer_excl_lung_and_haem", "haematological_cancer", "bone_marrow_transplan
t",
    "cystic_fibrosis", "sickle_cell_disease", "permanant_immunosuppression",
    "temporary_immunosuppression", "asplenia"]:
    df[c] = np.where(df[c]==1, "yes", "no")

df = df.rename(columns={"imd": "Index_of_Multiple_Deprivation", "ageband_community": "community_ageband"
})
# get total population sizes and names for each STP
stps = pd.read_csv(os.path.join("../", "lib", "stp_dict.csv"), usecols=["stp_id", "name", "list_size_o80"])
df = df.merge(stps, left_on="stp", right_on="stp_id", how="left").drop(["care_home_type", "age", "stp_i
d"], 1).rename(columns={"name": "stp_name"})

```

```

/root/.pyenv/versions/3.8.1/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3062: DtypeWa
rning: Columns (42) have mixed types.Specify dtype option on import or set low_memory=False.
    has_raised = await self.run_ast_nodes(code_ast.body, cell_name,

```

```

In [3]: # Find latest date of covid vaccinations covered in the dataset

latest_date = df[df["covid_vacc_date"]!=0]["covid_vacc_date"].max()
latest_date_fmt = datetime.strptime(latest_date, "%Y-%m-%d").strftime("%d %b %Y")
print("latest date ", latest_date_fmt)

```

```
latest date 13 Jan 2021
```

**Process data - summarise by group and demographics at latest date**

Calculate cumulative sums at each date and select latest date + previous figures for comparison

```

In [5]: # population subgroups - in a dict to indicate which field to filter on
pops = {"80+": "community_ageband",
        "70-79": "community_ageband",
        "care home": "community_ageband",
        "presumed healthcare workers (under 70, non-care home residents)": "other"
        }
groups = pops.keys()

def cumsums(df, groups):
    """
    Calculate cumulative sums

    Inputs:
    df (dataframe): input data
    groups(list): subgroups to breakdown by

    out:
    out2(dataframe)
    """
    df_dict_out = {}
    for k in groups:
        # filter dataframe to eligible group
        if pops[k] == "other": # for "all others" filter out the defined groups above
            out = df.copy().loc[(~df["community_ageband"].isin(["care home", "70-79", "80+"]))]
            cols = ["sex", "ageband", "Ethnicity_broad_categories", "Ethnicity_sixteen_categories", "Index_of_Multiple_Deprivation",
                    "bmi", "chronic_cardiac_disease", "current_copd", "dialysis", "dmards", "ssri"]
        elif k == "care home":
            out = df.copy().loc[(df[pops[k]] == k)]
            cols = ["sex", "ageband", "Ethnicity_broad_categories"]
        else: # age groups
            out = df.copy().loc[(df[pops[k]] == k)]
            cols = ["sex", "Ethnicity_broad_categories", "Ethnicity_sixteen_categories", "Index_of_Multiple_Deprivation",
                    "bmi", "chronic_cardiac_disease", "current_copd", "dialysis", "dmards", "dementia",
                    "psychosis_schiz_bipolar", "intel_dis_incl_downs_syndrome", "ssri",
                    "chemo_or_radio", "lung_cancer", "cancer_excl_lung_and_haem", "haematological_cancer"]

        df_dict_temp = {}

        # overall figures
        total = out[["patient_id"]].nunique()[0]
        out2 = out.copy().loc[(out["covid_vacc_flag"] == "vaccinated")]
        # group by date of covid vaccines to calculate cumulative sum of vaccines at each date of the campaign
        out2 = pd.DataFrame(out2.groupby(["covid_vacc_date"])[["patient_id"]].nunique().unstack().fillna(0).cumsum()).reset_index()
        out2 = out2.rename(columns={0: "overall"}).drop(["level_0"], 1)

        # in case no vaccinations on latest date for some STPs/groups, insert the latest data as a new row with the required date:
        if out2["covid_vacc_date"].max() < latest_date:
            out2.loc[max(out2.index)+1] = [latest_date, out2["overall"].max()]

        # suppress low numbers
        out2["overall"] = out2["overall"].replace([1, 2, 3, 4, 5, 6], 0).fillna(0).astype(int)
        out2["overall_total"] = total.astype(int)

        out2["overall_percent"] = 100*(out2["overall"]/out2["overall_total"])
        df_dict_temp["overall"] = out2.set_index("covid_vacc_date")

        # figures by demographic/clinical features
        for c in cols:
            if c == "sex":
                out = out.loc[out[c].isin(["M", "F"])]

            # find total number of patients in each subgroup (e.g. no of males and no of females)
            totals = out.groupby([c])[["patient_id"]].nunique().rename(columns={"patient_id": "total"}).transpose()

            # find total number of patients vaccinated in each subgroup (e.g. no of males and no of females),
            # cumulative at each date of the campaign
            out2 = out.copy().loc[(out["covid_vacc_flag"] == "vaccinated")]
            out2 = out2.groupby([c, "covid_vacc_date"])[["patient_id"]].nunique().unstack()
            out2 = out2.fillna(0).cumsum()

            # suppress low numbers

```

```
out2 = out2.replace([1,2,3,4,5,6], 0).fillna(0)
# round other values to nearest 7
out2 = 7*round(out2/7, 0)

for c2 in out2.columns:
    out2[f"{c2}_total"] = totals[c2][0].astype(int)
    #calculate percentage
    out2[f"{c2}_percent"] = 100*(out2[c2]/out2[f"{c2}_total"])

    # in case no vaccinations on latest date for some STPs/groups, insert the latest data as a
    new row with the required date:
    if out2.index.max() < latest_date:
        out2.loc[latest_date] = out2.max()

    df_dict_temp[c] = out2

    df_dict_out[k] = df_dict_temp

return df_dict_out

df_dict_cum = cumsums(df, groups)
```

```

In [6]: def report_results(df_dict_cum, group, breakdown=[]):
    '''
        Summarise data at latest date, overall and by demographic/clinical features, and including change
        from previous week.
        Processes one group (e.g. 80+) at a time so must be run within a loop to cover all required group
        s.

        Inputs:
        df_dict_cum (dict): dictionary of cumulative sums
        group (str): e.g. "80+" (one of first level index of df_dict_cum)
        breakdown (list):

        Outputs:
        out3 (Dataframe): summary data
    '''

    out = pd.DataFrame()
    out3 = pd.DataFrame()

    k=group

    if len(breakdown)== 0:
        breakdown = df_dict_cum[k].keys()

    for c in breakdown:
        out = df_dict_cum[k][c]

        # calculate changes: select only latest date and 7 days ago:
        latest = pd.to_datetime(out.index).max()
        lastweek = (latest + pd.DateOffset(days=-7)).strftime("%Y-%m-%d")
        lastweek = str(max(lastweek, out.index.min()))

        # filter to required values:

        # for groups with a population denominator, keep the percentage value only
        if k != "presumed healthcare workers (under 70, non-care home residents)":
            out = out.filter(regex='percent').round(1)
            col_str = " (percent)"
        # for groups with no denominator, keep the actual values only
        elif k == "presumed healthcare workers (under 70, non-care home residents)":
            # totals and percent not needed
            out = out.filter(regex='^(?!.*total).*$')
            out = out.filter(regex='^(?!.*percent).*$')
            col_str = ""

        # if last week's exact date not present, fill in using latest values prior to required date
        if sum(out.index == lastweek)==0:
            out.loc[lastweek] = out.loc[out.index < lastweek].max()

        out = out.loc[[latest_date,lastweek],:].transpose()

        '''# if there is only one non-zero value, round the remaining values to avoid suppressed value
        s being calculated
        for i in out2.index:
            if (out2.loc[i] == 0).astype(int).sum() == 1:
                out2.loc[i]=10*(out2.loc[i]/10).round(0)
            display(out2)'''

        out["weeklyrate"] = ((out[latest_date] - out[lastweek]).fillna(0)).round(1)
        out["Increase in uptake (%)"] = (100*(out["weeklyrate"]/out[lastweek]).fillna(0)).round(1)
        out["weeks_to_target"] = (90 - out[latest_date])/out["weeklyrate"]

        date_reached=pd.Series(dtype="datetime64[ns]", name="date_reached")

        for i in out.index:
            weeks_to_target = out["weeks_to_target"][i]

            if weeks_to_target <25: # if 6mo+ until expected to reach target, assume too little data t
o tell
                date_reached[i] = (latest + pd.DateOffset(days=weeks_to_target*7)).strftime('%d-%b')
            else:
                date_reached[i] = "unknown"
            out = out.transpose().append(date_reached).transpose().drop("weeks_to_target",1)
            out = out[[lastweek,"weeklyrate","date_reached","Increase in uptake (%)"]].rename(columns={las
tweek: f"vaccinated at {lastweek}{col_str}", "weeklyrate":f"Uptake over last 7d{col_str}", "date_reach
ed":f"Date projected to reach 90%"})
            out.index = out.index.str.replace("_percent", "")
            out = out.reset_index().rename(columns={c:"group", "index":"group"})
            out["category"] = c
            out = out.set_index(["category", "group"])

```

```

##### n, percent and total pop figures for latest date
out2 = df_dict_cum[k][c].reset_index()
out2 = out2.loc[out2["covid_vacc_date"]==latest_date].reset_index().set_index("covid_vacc_date").drop(["index"], 1).transpose()
# split field names e.g. "M_percent" -> "M" "percent"
out2.index = pd.MultiIndex.from_tuples(out2.index.str.split('_').tolist())

out2 = out2.unstack().reset_index(col_level=1)
out2.columns = out2.columns.droplevel()
out2 = out2.rename(columns={"index": "group", np.nan: "vaccinated"})
out2["percent"] = out2["percent"].round(1)
out2["category"] = c
out2 = out2.set_index(["category", "group"])

out2 = out2.join(out)

out3 = out3.append(out2)

if k == "presumed healthcare workers (under 70, non-care home residents)":
    out3 = out3.drop(["percent", "total", "Date projected to reach 90%"], 1)
else:
    out3 = out3.drop(["Increase in uptake (%)"], 1)

return out3

```

## Cumulative vaccination figures - overall

```

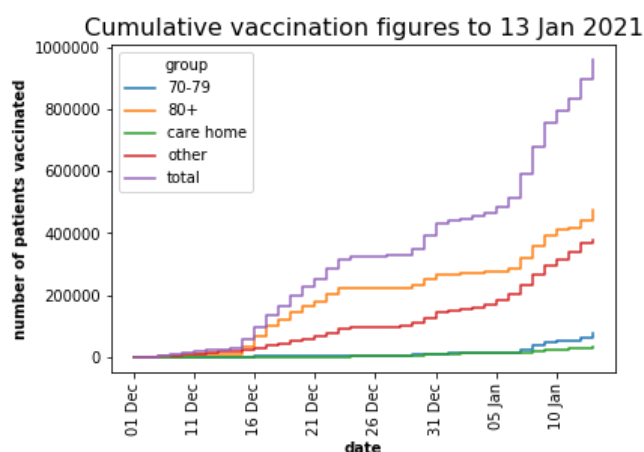
In [7]: # cumulative chart by day of total vaccines given
dfp = df.copy().loc[(df["covid_vacc_date"]!=0)]
dfp["group"] = np.where(dfp["community_ageband"].isin(["80+", "70-79", "care home"]), dfp["community_ageband"], "other")
dfp = dfp.groupby(["covid_vacc_date", "group"])[["patient_id"]].count()
dfp = dfp.unstack().fillna(0).cumsum().reset_index().replace([0,1,2,3,4,5],0)

dfp["covid_vacc_date"] = pd.to_datetime(dfp["covid_vacc_date"]).dt.strftime("%d %b")
dfp = dfp.set_index("covid_vacc_date")
dfp.columns = dfp.columns.droplevel()
dfp["total"] = dfp.sum(axis=1)

dfp.plot(legend=True, ds='steps-post')

plt.xlabel("date", fontweight='bold')
plt.xticks(rotation=90)
plt.ylabel("number of patients vaccinated", fontweight='bold')
plt.title(f"Cumulative vaccination figures to {latest_date_fmt}", fontsize=16)
plt.show()

```



```

In [8]: # summarise data at latest date
groups = ["80+", "70-79", "care home", "presumed healthcare workers (under 70, non-care home residents)"]
df_dict_latest = {}
for g in groups:
    out = report_results(df_dict_cum, g)
    df_dict_latest[g] = out

```

## Proportion of each eligible population vaccinated to date

```
In [9]: display(Markdown(f"### As at {latest_date_fmt}"))

def round7(n):
    '''Round n to nearest 7 and output as integer'''
    return ( int(7*round((n/7),1)) )

vaccinated_total = round7( df.loc[df["covid_vacc_date"]!=0]["patient_id"].nunique() )
display(Markdown(f"**Total** population vaccinated in TPP {vaccinated_total:,d}"))

for k in groups:
    out = df_dict_latest[k]
    vaccinated = round7(out.loc[("overall", "overall")]["vaccinated"])
    if k != "presumed healthcare workers (under 70, non-care home residents)":
        percent = out.loc[("overall", "overall")]["percent"].round(1)
        total = out.loc[("overall", "overall")]["total"].astype(int)
        out_str = f"**{k}** population vaccinated {vaccinated:,} ({percent}% of {total:,})"
    else:
        out_str = f"**{k}** population vaccinated {vaccinated:,}"
    if k=="80+": # store values for later use
        overall_rate_80 = percent

    display(Markdown(out_str))

# count oxford vax as a proportion of total; filter to date of first vax only in case of patients having mixed types
oxford_vaccines = round7(df.copy().loc[df["covid_vacc_date"]==df["covid_vacc_oxford_date"]]["covid_vacc_c_flag_ox"].sum())
ox_percent = round(100*oxford_vaccines/vaccinated_total, 1)
second_doses = round7(df["covid_vacc_2nd"].sum())
sd_percent = round(100*second_doses/vaccinated_total, 1)

display(Markdown(f"#### Vaccine type and second doses"),
        Markdown(f"- Of those vaccinated, {second_doses:,} have received a **second dose** ({sd_percent}%)" ),
        Markdown(f"- Of those vaccinated, {oxford_vaccines:,} received the **Oxford vaccine** as their first dose ({ox_percent}%)" ),
        Markdown(f" "),
        Markdown("figures rounded to nearest 7"))
```

### As at 13 Jan 2021

**Total** population vaccinated in TPP 961,580

**80+** population vaccinated 476,375 (41.1% of 1,160,062)

**70-79** population vaccinated 74,108 (3.6% of 2,080,656)

**care home** population vaccinated 32,174 (33.2% of 96,997)

**presumed healthcare workers (under 70, non-care home residents)** population vaccinated 378,921

#### Vaccine type and second doses

- Of those vaccinated, 169,472 have received a **second dose** (17.6%)
- Of those vaccinated, 65,055 received the **Oxford vaccine** as their first dose (6.8%)

figures rounded to nearest 7

## Detailed summary of uptake among population groups as at latest date



```
In [10]: pd.set_option('display.max_rows',200)

for g in groups:
    display(Markdown(f"## "),
             Markdown(f"## COVID vaccination rollout among **{g}** population up to {latest_date_fmt}"
    ),
             Markdown(f"- 'Date projected to reach 90%' being 'unknown' indicates projection of >6mo (l
likely insufficient information)\n"
                    f"- Patient counts rounded to the nearest 7"))
    out = df_dict_latest[g]
    # suppress previous week's values if change in vaccinated number is <10
    ''' if g!="presumed healthcare workers (under 70, non-care home residents)":
        out["Uptake over Last 7d (percent)"]=np.where(out["Uptake over Last 7d (percent)"]*out["tota
L"]>=1000,
                                                    out["Uptake over Last 7d (percent)"],0)
        out["vaccinated at 2020-12-30 (percent)"]=np.where(out["Uptake over Last 7d (percent)"]*out["t
otal"]>=1000,
                                                    out["vaccinated at 2020-12-30 (percent)"],
0)'''
    display(out)
```

## COVID vaccination rollout among 80+ population up to 13 Jan 2021

- 'Date projected to reach 90%' being 'unknown' indicates projection of >6mo (likely insufficient information)
- Patient counts rounded to the nearest 7

		vaccinated	percent	total	vaccinated at 2021-01-06 (percent)	Uptake over last 7d (percent)	Date projected to reach 90%
category	group						
overall	overall	476376.0	41.1	1160062.0	24.5	16.6	02-Feb
sex	F	265076.0	39.6	669278.0	23.2	16.4	03-Feb
	M	211295.0	43.1	490774.0	26.3	16.8	01-Feb
Ethnicity_broad_categories	Black	2121.0	20.5	10329.0	11.6	8.9	08-Mar
	Mixed	756.0	27.0	2805.0	16.2	10.8	22-Feb
	Other	1498.0	27.0	5539.0	15.8	11.2	21-Feb
	South Asian	7945.0	29.5	26936.0	17.7	11.8	17-Feb
	Unknown	129199.0	39.7	325637.0	23.3	16.4	03-Feb
	White	334852.0	42.5	788806.0	25.5	17	01-Feb
	White	334852.0	42.5	788806.0	25.5	17	01-Feb
Ethnicity_sixteen_categories	African	329.0	15.9	2072.0	9.5	6.4	04-Apr
	Bangladeshi or British Bangladeshi	287.0	23.0	1250.0	12.3	10.7	25-Feb
	Caribbean	1533.0	21.8	7045.0	12.1	9.7	03-Mar
	Chinese	497.0	31.9	1558.0	18.9	13	13-Feb
	Other	1001.0	25.1	3981.0	14.6	10.5	25-Feb
	Other Asian	1204.0	28.8	4179.0	16.4	12.4	16-Feb
	British or Mixed British	317037.0	42.8	740996.0	25.7	17.1	01-Feb
	Indian or British Indian	4781.0	33.8	14159.0	20.9	12.9	12-Feb
	Irish	3346.0	41.2	8126.0	25.2	16	03-Feb
	Other Black	252.0	20.8	1212.0	12.1	8.7	09-Mar
	Other White	14462.0	36.4	39684.0	21.1	15.3	06-Feb
	Other mixed	259.0	30.9	839.0	19.2	11.7	17-Feb
	Pakistani or British Pakistani	1673.0	22.8	7348.0	13.3	9.5	03-Mar
	Unknown	129199.0	39.7	325637.0	23.3	16.4	03-Feb
	White + Asian	168.0	35.7	471.0	23.8	11.9	13-Feb
	White + Black African	77.0	20.9	368.0	13.3	7.6	17-Mar
	White + Black Caribbean	252.0	22.4	1127.0	11.8	10.6	26-Feb
Index_of_Multiple_Deprivation	1 Most deprived	57393.0	37.9	151255.0	22.8	15.1	06-Feb
	2	76223.0	39.2	194657.0	23.8	15.4	05-Feb
	3	100996.0	39.0	259041.0	23.2	15.8	04-Feb
	4	112749.0	42.2	266905.0	25.3	16.9	01-Feb
	5 Least deprived	119567.0	44.7	267454.0	26.1	18.6	30-Jan
	Unknown	9443.0	45.5	20740.0	29.2	16.3	01-Feb
	Unknown	9443.0	45.5	20740.0	29.2	16.3	01-Feb
bmi	30+	84000.0	41.2	203700.0	24.8	16.4	02-Feb
	under 30	392371.0	41.0	956352.0	24.4	16.6	02-Feb
chronic_cardiac_disease	no	333459.0	40.9	814701.0	24.2	16.7	02-Feb
	yes	142912.0	41.4	345351.0	25.1	16.3	02-Feb
current_copd	no	424641.0	40.9	1037745.0	24.3	16.6	02-Feb
	yes	51730.0	42.3	122307.0	26.2	16.1	02-Feb
dialysis	no	475482.0	41.1	1157832.0	24.5	16.6	02-Feb
	yes	889.0	40.0	2220.0	25.5	14.5	06-Feb
dmards	no	461223.0	41.0	1124310.0	24.4	16.6	02-Feb
	yes	15148.0	42.4	35742.0	26.2	16.2	02-Feb
dementia	no	470274.0	41.2	1140315.0	24.6	16.6	02-Feb
	yes	6097.0	30.9	19737.0	16.5	14.4	10-Feb
psychosis_schiz_bipolar	no	473620.0	41.2	1150959.0	24.6	16.6	02-Feb

		population_characteristics			vaccinated at 2021-01-06 (percent)	Uptake over last 7d (percent)	Date projected to reach 90%
category	group	vaccinated	percent	total			
intel_dis_incl_downs_syndrome	yes	2751.0	30.3	9093.0	16.9	13.4	13-Feb
	no	475874.0	41.1	1158286.0	24.5	16.6	02-Feb
ssri	yes	497.0	28.1	1766.0	15.1	13	15-Feb
	no	444199.0	41.3	1076491.0	24.6	16.7	02-Feb
chemo_or_radio	yes	32172.0	38.5	83561.0	22.7	15.8	04-Feb
	no	459634.0	41.0	1121226.0	24.4	16.6	02-Feb
lung_cancer	yes	16737.0	43.1	38826.0	26.5	16.6	01-Feb
	no	473186.0	41.1	1152173.0	24.5	16.6	02-Feb
cancer_excl_lung_and_haem	yes	3185.0	40.4	7879.0	25.6	14.8	05-Feb
	no	387961.0	40.7	954235.0	24.1	16.6	02-Feb
haematological_cancer	yes	88410.0	43.0	205817.0	26.2	16.8	01-Feb
	no	467061.0	41.0	1138825.0	24.4	16.6	02-Feb
	yes	9310.0	43.9	21227.0	27.3	16.6	01-Feb

## COVID vaccination rollout among 70-79 population up to 13 Jan 2021

- 'Date projected to reach 90%' being 'unknown' indicates projection of >6mo (likely insufficient information)
- Patient counts rounded to the nearest 7

		vaccinated	percent	total	vaccinated at 2021-01-06 (percent)	Uptake over last 7d (percent)	Date projected to reach 90%
category	group						
overall	overall	74108.0	3.6	2080656.0	0.7	2.9	unknown
sex	F	40124.0	3.7	1088308.0	0.8	2.9	unknown
	M	33985.0	3.4	992341.0	0.6	2.8	unknown
Ethnicity_broad_categories	Black	434.0	3.2	13635.0	0.6	2.6	unknown
	Mixed	196.0	3.5	5573.0	0.8	2.7	unknown
	Other	399.0	3.1	12850.0	0.7	2.4	unknown
	South Asian	2660.0	5.2	50841.0	1.7	3.5	01-Jul
	Unknown	15071.0	3.3	462097.0	0.6	2.7	unknown
	White	55356.0	3.6	1535653.0	0.7	2.9	unknown
	White + Asian	42.0	3.5	1190.0	0.6	2.9	unknown
Ethnicity_sixteen_categories	African	154.0	3.0	5122.0	0.8	2.2	unknown
	Bangladeshi or British Bangladeshi	84.0	5.1	1659.0	0.8	4.3	31-May
	Caribbean	217.0	3.4	6297.0	0.6	2.8	unknown
	Chinese	98.0	2.7	3642.0	0.4	2.3	unknown
	Other	294.0	3.2	9208.0	0.8	2.4	unknown
	Other Asian	427.0	4.1	10378.0	1.3	2.8	unknown
	British or Mixed British	51709.0	3.6	1445140.0	0.7	2.9	unknown
	Indian or British Indian	1519.0	5.4	28019.0	1.9	3.5	01-Jul
	Irish	784.0	5.9	13361.0	1.3	4.6	20-May
	Other Black	63.0	2.8	2216.0	0.6	2.2	unknown
	Other White	2856.0	3.7	77152.0	0.8	2.9	unknown
	Other mixed	56.0	2.8	2003.0	0.7	2.1	unknown
	Pakistani or British Pakistani	630.0	5.8	10785.0	1.6	4.2	02-Jun
	Unknown	15071.0	3.3	462097.0	0.6	2.7	unknown
	White + Asian	42.0	3.5	1190.0	0.6	2.9	unknown
Index_of_Multiple_Deprivation	White + Black African	35.0	3.4	1016.0	1.4	2	unknown
	White + Black Caribbean	56.0	4.1	1364.0	0.5	3.6	29-Jun
	1 Most deprived	12397.0	4.5	276568.0	0.9	3.6	28-Jun
	2	12852.0	3.6	353767.0	0.7	2.9	unknown
	3	14637.0	3.2	463499.0	0.6	2.6	unknown
	4	17710.0	3.7	478790.0	0.8	2.9	unknown
	5 Least deprived	14931.0	3.2	470838.0	0.6	2.6	unknown
bmi	Unknown	1582.0	4.3	37187.0	0.7	3.6	28-Jun
	30+	17927.0	3.5	519156.0	0.6	2.9	unknown
	under 30	56182.0	3.6	1561493.0	0.7	2.9	unknown
chronic_cardiac_disease	no	58814.0	3.4	1711422.0	0.7	2.7	unknown
	yes	15295.0	4.1	369227.0	0.8	3.3	unknown
current_copd	no	65646.0	3.5	1882694.0	0.7	2.8	unknown
	yes	8463.0	4.3	197955.0	0.8	3.5	03-Jul
dialysis	no	73927.0	3.6	2076503.0	0.7	2.9	unknown
	yes	182.0	4.4	4146.0	0.7	3.7	23-Jun
dmards	no	71281.0	3.5	2008604.0	0.7	2.8	unknown
	yes	2828.0	3.9	72045.0	0.8	3.1	unknown
dementia	no	73633.0	3.6	2072820.0	0.7	2.9	unknown
	yes	476.0	6.1	7829.0	1.3	4.8	15-May
psychosis_schiz_bipolar	no	73325.0	3.6	2059466.0	0.7	2.9	unknown

		population_characteristics			vaccinated at 2021-01-06 (percent)	Uptake over last 7d (percent)	Date projected to reach 90%
category	group	vaccinated	percent	total			
intel_dis_incl_downs_syndrome	yes	784.0	3.7	21183.0	0.8	2.9	unknown
	no	73829.0	3.6	2073921.0	0.7	2.9	unknown
ssri	yes	280.0	4.2	6728.0	1.1	3.1	unknown
	no	68054.0	3.5	1917148.0	0.7	2.8	unknown
chemo_or_radio	yes	6055.0	3.7	163501.0	0.7	3	unknown
	no	71491.0	3.6	2011462.0	0.7	2.9	unknown
lung_cancer	yes	2618.0	3.8	69187.0	0.7	3.1	unknown
	no	73556.0	3.6	2068011.0	0.7	2.9	unknown
cancer_excl_lung_and_haem	yes	553.0	4.4	12638.0	0.8	3.6	28-Jun
	no	62888.0	3.5	1796838.0	0.7	2.8	unknown
haematological_cancer	yes	11221.0	4.0	283811.0	0.8	3.2	unknown
	no	72842.0	3.6	2050414.0	0.7	2.9	unknown
	yes	1267.0	4.2	30235.0	0.7	3.5	03-Jul

## COVID vaccination rollout among care home population up to 13 Jan 2021

- 'Date projected to reach 90%' being 'unknown' indicates projection of >6mo (likely insufficient information)
- Patient counts rounded to the nearest 7

		vaccinated	percent	total	vaccinated at 2021-01-06 (percent)	Uptake over last 7d (percent)	Date projected to reach 90%
category	group						
overall	overall	32174.0	33.2	96997.0	12.6	20.6	01-Feb
sex	F	23149.0	33.8	68572.0	12.8	21	31-Jan
	M	9023.0	31.7	28424.0	12.3	19.4	03-Feb
ageband	65-69	966.0	28.4	3404.0	11.3	17.1	07-Feb
	70-79	5523.0	31.8	17356.0	12.3	19.5	02-Feb
	80+	25690.0	33.7	76236.0	12.8	20.9	31-Jan
Ethnicity_broad_categories	Black	98.0	21.0	467.0	3	18	08-Feb
	Mixed	49.0	24.4	201.0	7	17.4	08-Feb
	Other	84.0	25.1	334.0	10.5	14.6	13-Feb
	South Asian	168.0	27.1	621.0	6.8	20.3	03-Feb
	Unknown	7721.0	34.1	22620.0	13.5	20.6	31-Jan
	White	24052.0	33.1	72753.0	12.5	20.6	01-Feb

## COVID vaccination rollout among presumed healthcare workers (under 70, non-care home residents) population up to 13 Jan 2021

- 'Date projected to reach 90%' being 'unknown' indicates projection of >6mo (likely insufficient information)
- Patient counts rounded to the nearest 7

		vaccinated	vaccinated at 2021-01-06	Uptake over last 7d	Increase in uptake (%)
category	group				
overall	overall	378922.0	202846	176076	86.8
sex	F	284669.0	150829	133840	88.7
	M	94248.0	52017	42231	81.2
ageband	0-19	3269.0	1512	1757	116.2
	20-29	46928.0	23534	23394	99.4
	30-39	72156.0	38157	33999	89.1
	40-49	87948.0	48069	39879	83
	50-59	111223.0	60361	50862	84.3
	60-69	57393.0	31213	26180	83.9
Ethnicity_broad_categories	Black	6468.0	3339	3129	93.7
	Mixed	4249.0	2422	1827	75.4
	Other	7301.0	4340	2961	68.2
	South Asian	31276.0	18998	12278	64.6
	Unknown	84658.0	45675	38983	85.3
	White	244965.0	128072	116893	91.3
Ethnicity_sixteen_categories	African	4333.0	2268	2065	91
	Bangladeshi or British Bangladeshi	1015.0	539	476	88.3
	Caribbean	973.0	476	497	104.4
	Chinese	1596.0	959	637	66.4
	Other	5705.0	3381	2324	68.7
	Other Asian	9625.0	6027	3598	59.7
	British or Mixed British	225085.0	117383	107702	91.8
	Indian or British Indian	15722.0	9681	6041	62.4
	Irish	2016.0	1148	868	75.6
	Other Black	1162.0	595	567	95.3
	Other White	17871.0	9548	8323	87.2
	Other mixed	1491.0	875	616	70.4
	Pakistani or British Pakistani	4914.0	2758	2156	78.2
	Unknown	84658.0	45675	38983	85.3
	White + Asian	1281.0	749	532	71
	White + Black African	819.0	476	343	72.1
	White + Black Caribbean	658.0	322	336	104.3
Index_of_Multiple_Deprivation	1 Most deprived	58135.0	31024	27111	87.4
	2	69993.0	37142	32851	88.4
	3	79457.0	42028	37429	89.1
	4	81424.0	43883	37541	85.5
	5 Least deprived	79891.0	43400	36491	84.1
	Unknown	10017.0	5376	4641	86.3
bmi	30+	82915.0	43967	38948	88.6
	under 30	296002.0	158879	137123	86.3
chronic_cardiac_disease	no	367941.0	196770	171171	87
	yes	10976.0	6076	4900	80.6
current_copd	no	373716.0	200046	173670	86.8
	yes	5201.0	2800	2401	85.8
dialysis	no	378532.0	202622	175910	86.8
	yes	385.0	224	161	71.9
dmards	no	370510.0	197813	172697	87.3

category	group	vaccinated	vaccinated at 2021-01-06	Uptake over last 7d	Increase in uptake (%)
ssri	yes	8407.0	5033	3374	67
	no	320369.0	172312	148057	85.9
	yes	58548.0	30534	28014	91.7

Demographics time trend charts



```

In [11]: def plot_dem_charts(df_dict, include_overall=False):
    if include_overall==False:
        groups = ["sex","Ethnicity_broad_categories","Index_of_Multiple_Deprivation", "bmi", "chronic_
cardiac_disease", "current_copd", "dialysis", "dmards",
                  "psychosis_schiz_bipolar","intel_dis_incl_downs_syndrome","dementia", "ssri",
                  "chemo_or_radio", "lung_cancer", "cancer_excl_lung_and_haem", "haematological_can
cer"]
    else:
        groups = ["overall","sex","Ethnicity_broad_categories","Index_of_Multiple_Deprivation", "bmi",
"chronic_cardiac_disease", "current_copd", "dialysis", "dmards",
                  "psychosis_schiz_bipolar","intel_dis_incl_downs_syndrome","dementia", "ssri",
                  "chemo_or_radio", "lung_cancer", "cancer_excl_lung_and_haem", "haematological_can
cer"]

    for k in ["80+"]:
        display(Markdown(f"## COVID vaccination rollout among **{k}** population up to {latest_date_fm
t}"))

        out=df_dict[k]
        for c in groups:
            out=df_dict[k][c]
            # suppress low numbers
            cols = out.columns[(~out.columns.str.contains("_percent")) & (~out.columns.str.contains("_
total"))]
            for c2 in cols:
                # drop vaccinated and total column but keep percentage
                out = out.drop([c2, f"{c2}_total"],1)
                out = out.rename(columns={f"{c2}_percent":c2})

            display(Markdown(f"### COVID vaccinations among **{k}** population by **{c.replace('_', '
')}**"))
            out = out.reset_index()
            out["covid_vacc_date"] = pd.to_datetime(out["covid_vacc_date"]).dt.strftime("%d %b")
            out = out.set_index("covid_vacc_date")

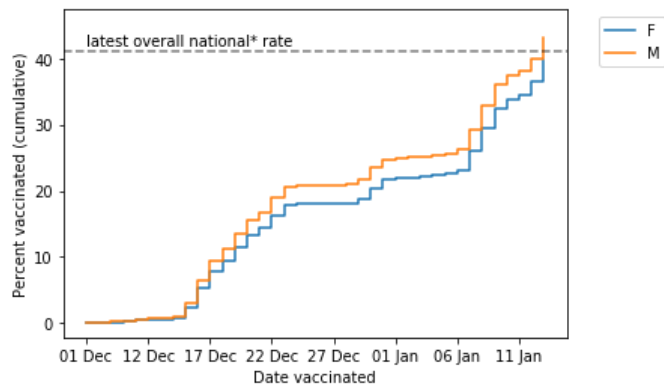
            out.plot(legend=True, ds='steps-post')
            plt.axhline(overall_rate_80, color="k", linestyle="--", alpha=0.5)
            plt.text(0, overall_rate_80*1.02, "latest overall national* rate")
            plt.ylim(top=1.1*max(overall_rate_80, out.max().max()))
            plt.ylabel("Percent vaccinated (cumulative)")
            plt.xlabel("Date vaccinated")
            plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
            plt.show()

plot_dem_charts(df_dict_cum)

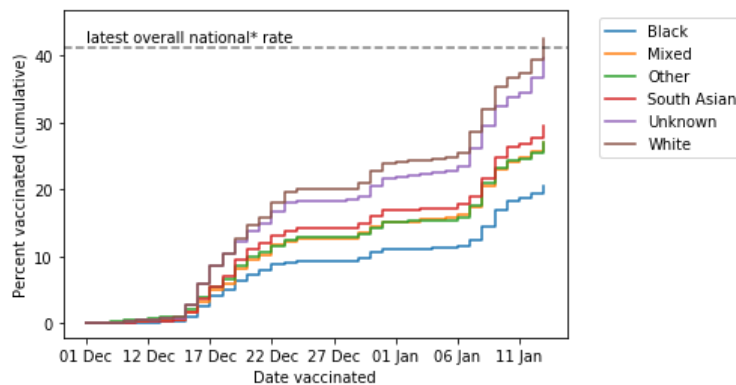
```

## COVID vaccination rollout among 80+ population up to 13 Jan 2021

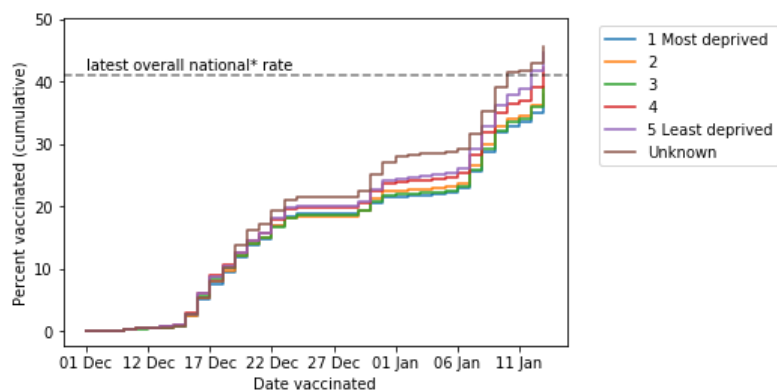
### COVID vaccinations among 80+ population by sex



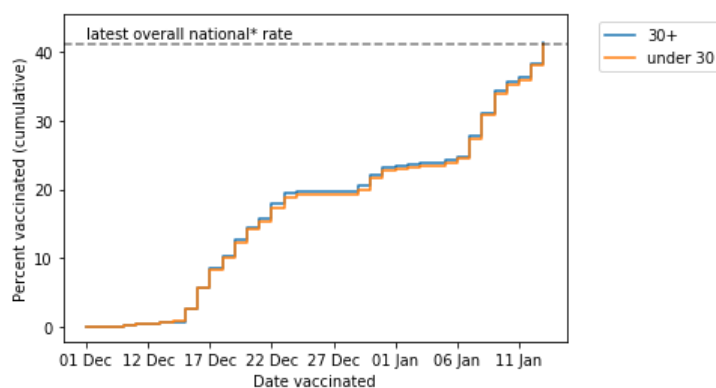
### COVID vaccinations among 80+ population by Ethnicity broad categories



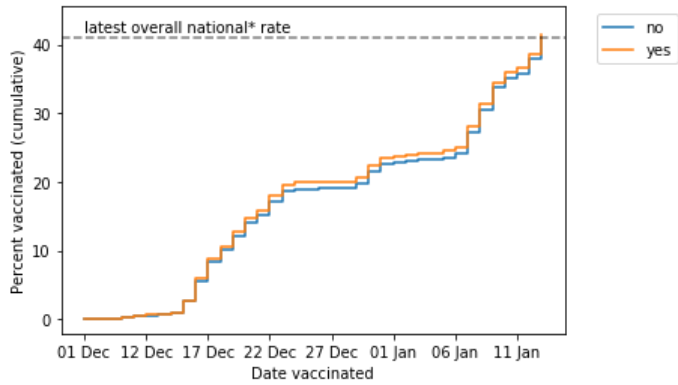
### COVID vaccinations among 80+ population by Index of Multiple Deprivation



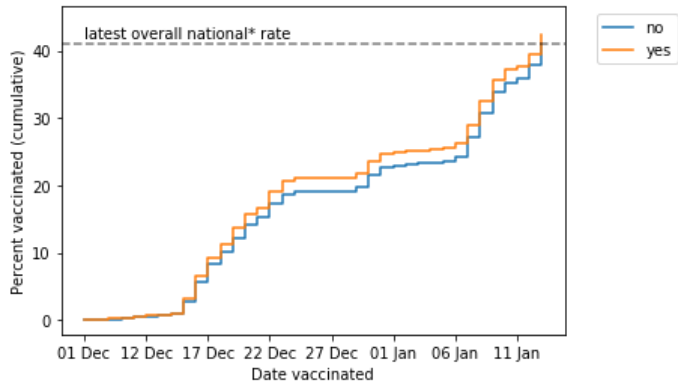
### COVID vaccinations among 80+ population by bmi



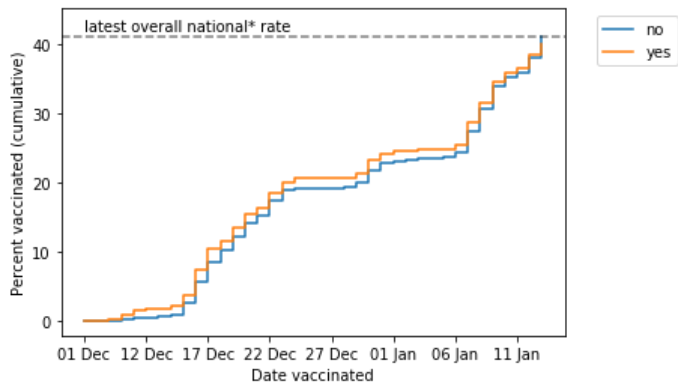
### COVID vaccinations among 80+ population by chronic cardiac disease



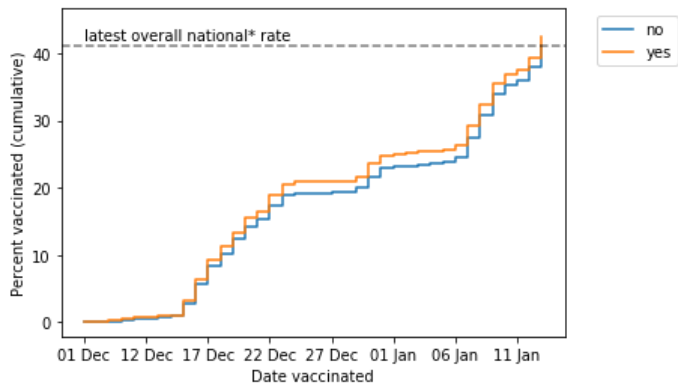
COVID vaccinations among 80+ population by current copd



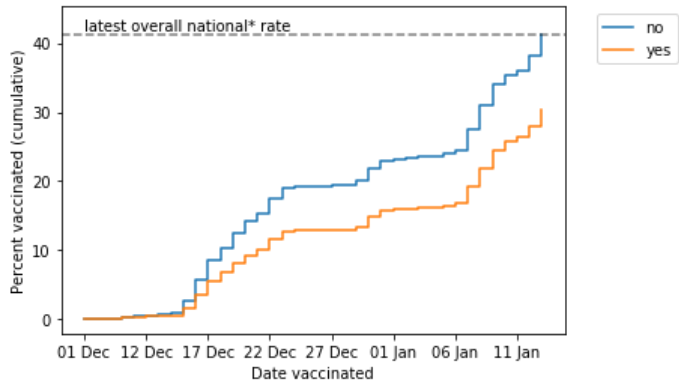
COVID vaccinations among 80+ population by dialysis



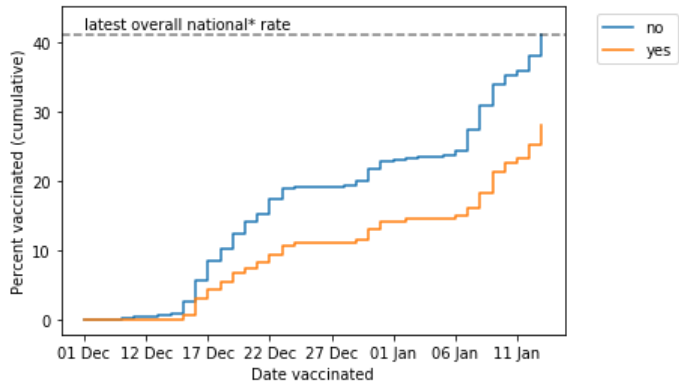
COVID vaccinations among 80+ population by dmards



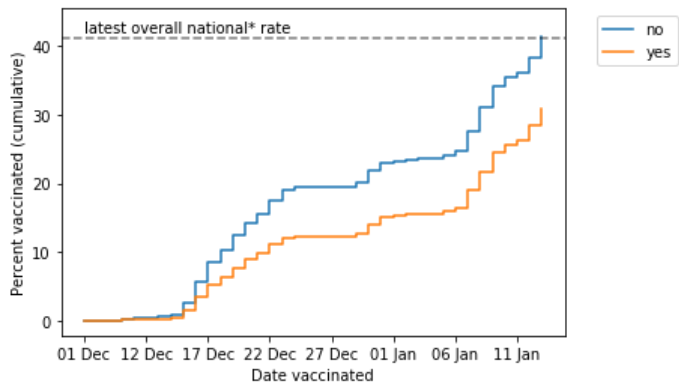
COVID vaccinations among 80+ population by psychosis schiz bipolar



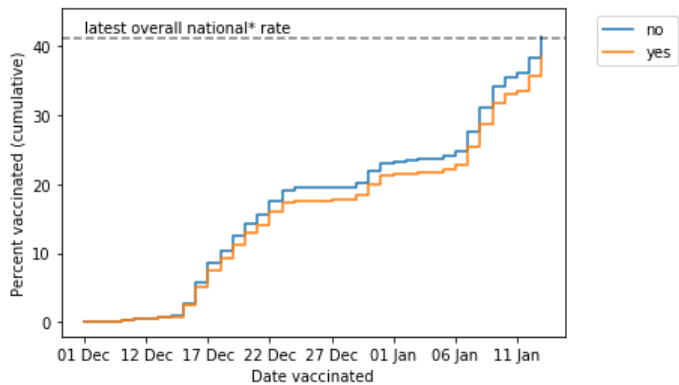
**COVID vaccinations among 80+ population by intel dis incl downs syndrome**



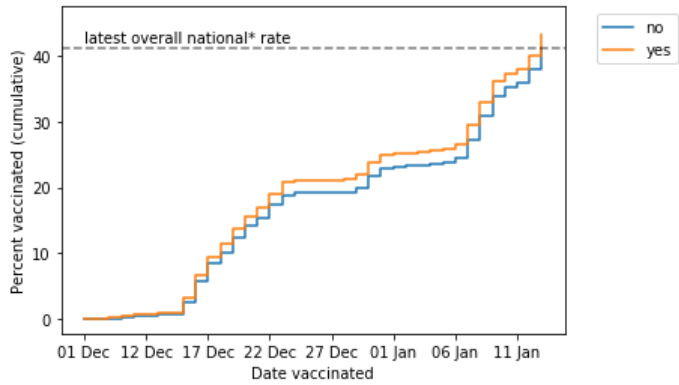
**COVID vaccinations among 80+ population by dementia**



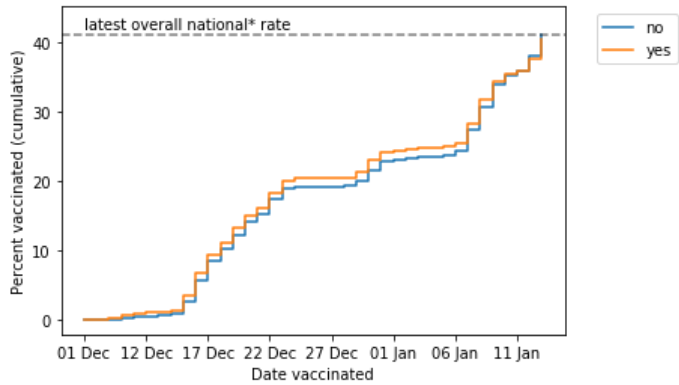
**COVID vaccinations among 80+ population by ssri**



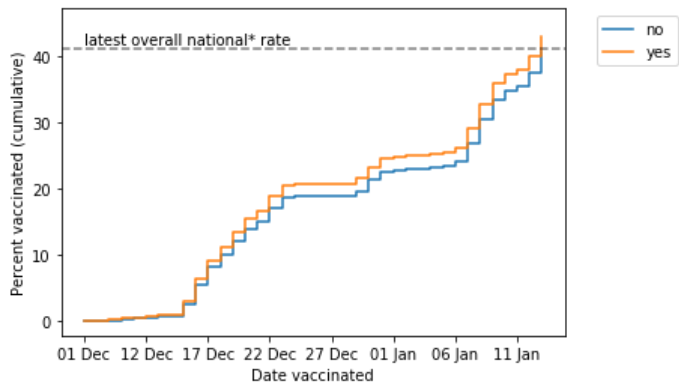
**COVID vaccinations among 80+ population by chemo or radio**



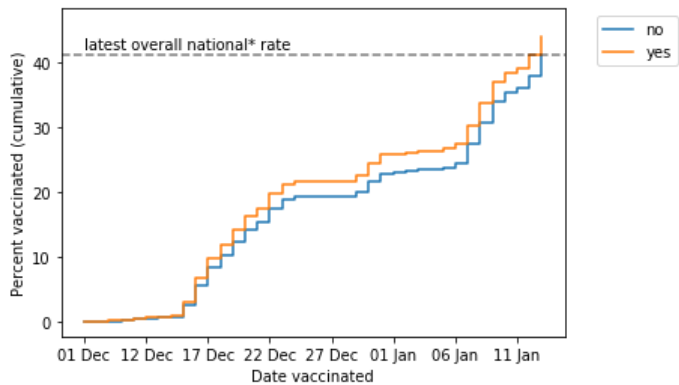
**COVID vaccinations among 80+ population by lung cancer**



**COVID vaccinations among 80+ population by cancer excl lung and haem**



**COVID vaccinations among 80+ population by haematological cancer**



**Completeness of ethnicity recording**

```

In [14]: # overall among vaccinated people
vaccinated = df[["covid_vacc_date", "community_ageband", "Ethnicity_broad_categories", "patient_id"]].loc[
    df["covid_vacc_date"] != 0]
vaccinated_ethnicity = vaccinated.groupby("Ethnicity_broad_categories")[["patient_id"]].nunique().reset_index()
out = vaccinated_ethnicity.loc[vaccinated_ethnicity["Ethnicity_broad_categories"] != "Unknown"]["patient_id"].sum() / \
    vaccinated_ethnicity["patient_id"].sum()

# in subgroups with denominators
for k in ["80+", "70-79", "care home"]:
    out = df[["community_ageband", "Ethnicity_broad_categories", "patient_id", "care_home"]].copy()
    out = out.loc[out[pops[k]] == k]

    total = out["patient_id"].nunique()

    known_eth = out.groupby("Ethnicity_broad_categories")[["patient_id"]].nunique().reset_index()
    known_eth = known_eth.loc[vaccinated_ethnicity["Ethnicity_broad_categories"] != "Unknown"]["patient_id"].sum()
    percent = round(100 * (known_eth / total), 1)
    display(Markdown(f"Total {k} population with ethnicity recorded {known_eth},d ({percent}%)"))

other = df[["community_ageband", "Ethnicity_broad_categories", "patient_id"]].copy()
other = other.loc[~other["community_ageband"].isin(["care home", "80+", "70-79"])]
vaccinated_ethnicity = vaccinated.groupby("Ethnicity_broad_categories")[["patient_id"]].nunique().reset_index()
out = vaccinated_ethnicity.loc[vaccinated_ethnicity["Ethnicity_broad_categories"] != "Unknown"]["patient_id"].sum() / \
    vaccinated_ethnicity["patient_id"].sum()
display(Markdown(f"Vaccinated presumed healthcare workers (under 70, non-care home residents) population with ethnicity recorded {other['patient_id'].nunique():,d ({percent}%)"))

```

Total **80+** population with ethnicity recorded 834,423 (71.9%)

Total **70-79** population with ethnicity recorded 1,618,557 (77.8%)

Total **care home** population with ethnicity recorded 74,377 (76.7%)

Vaccinated **presumed healthcare workers (under 70, non-care home residents)** population with ethnicity recorded 386,913 (76.7%)