CDC patient-level Covid data

November 9, 2020

Andy Feltovich

Import libraries

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
import datetime
```

---

import dataset

data and accompanying explanation available at:

https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/vbim-akqf

In [4]:
```
url = "https://data.cdc.gov/api/views/vbim-akqf/rows.csv?accessType=DOWNLOAD&bom=true&format=true"
```

In [5]:
```
# using abbreviation "ds" for "dataset"
ds = pd.read_csv(url, low_memory=False)
```

---

assess variable death_yn

metadata defines simply as "Death status"

In [6]:
```
print(ds["death_yn"].value_counts())
```
```
No         2662955
Missing    2321809
Unknown     617047
Yes         158255
Name: death_yn, dtype: int64
```

aren't sure whether or not the patient died?...very reassuring

---

assess death counts by age_group with crosstab

In [7]:
```
death_age_cross_tab = pd.crosstab(ds["age_group"], ds["death_yn"], margins=True)
```

format crosstab so it prints numbers > 999 with commas (e.g. 1,000) and print

In [8]:
```
death_age_cross_tab.style.format("{:,.0f}")
```

Out[8]:

death_yn age_group	Missing	No	Unknown	Yes	All
0 - 9 Years	71,676	100,117	17,582	55	189,430
10 - 19 Years	214,867	270,695	59,379	95	545,036
20 - 29 Years	463,612	533,316	130,201	720	1,127,849
30 - 39 Years	395,426	447,131	99,832	1,976	944,365
40 - 49 Years	365,776	413,671	93,516	4,798	877,761
50 - 59 Years	353,996	392,682	90,938	12,409	850,025
60 - 69 Years	239,445	267,322	62,428	26,292	595,487
70 - 79 Years	123,586	140,717	33,836	38,943	337,082
80+ Years	88,020	95,357	28,650	72,946	284,973
Unknown	5,368	1,903	683	12	7,966
All	2,321,772	2,662,911	617,045	158,246	5,759,974

calculate variable for probability of death by age_group: prob_death

In [9]:
```
death_age_cross_tab["prob_death"] = death_age_cross_tab["Yes"]/(death_age_cross_tab["All"])
```

calculate variable for odds of death by age_group: odds_death

In [10]:
```
death_age_cross_tab["odds_death"] = death_age_cross_tab["Yes"]/(death_age_cross_tab["All"]-death_age_cr
oss_tab["Yes"])
```

print new variables

In [11]:
```
print(death_age_cross_tab[["prob_death","odds_death"]])
```
```
death_yn       prob_death  odds_death
age_group
0 - 9 Years      0.000290    0.000290
10 - 19 Years    0.000174    0.000174
20 - 29 Years    0.000638    0.000639
30 - 39 Years    0.002092    0.002097
40 - 49 Years    0.005466    0.005496
50 - 59 Years    0.014598    0.014815
60 - 69 Years    0.044152    0.046192
70 - 79 Years    0.115530    0.130620
80+ Years        0.255975    0.344041
Unknown          0.001506    0.001509
All              0.027473    0.028249
```

drop rows for age_group = Unkown and age_group = All

In [12]:
```
death_age_cross_tab = death_age_cross_tab.drop(["Unknown", "All"])
```

calculate increase in odds of death for every 10 year increase in age

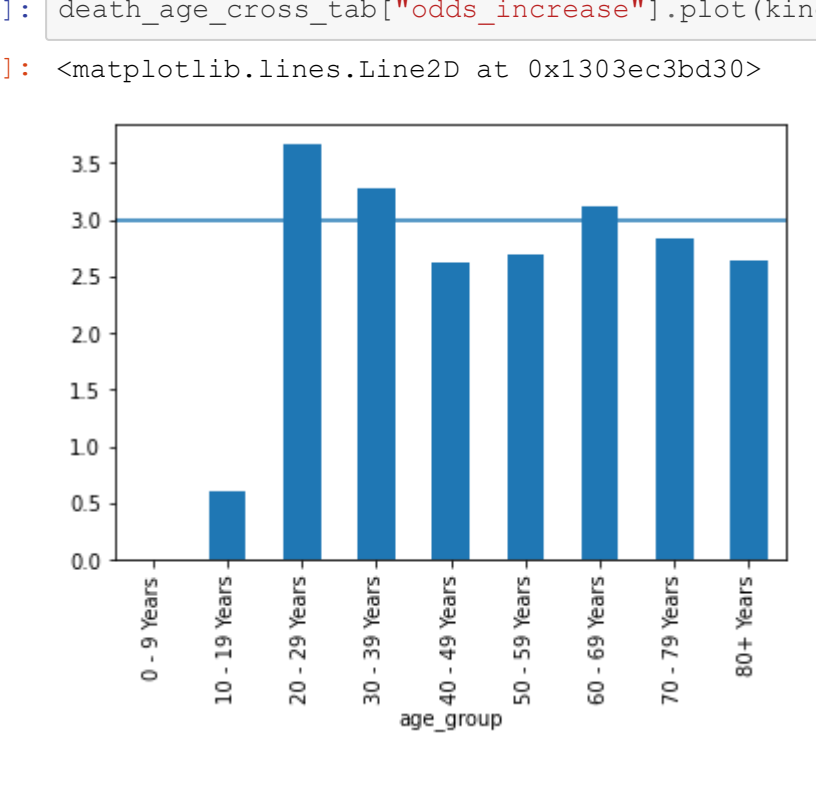
In [13]:
```
odds_temp = [None]
i = 1
for index, row in death_age_cross_tab.iterrows():
    if i > 1:
        odds_increase = row['odds_death']/previous_odds
        odds_temp.append(odds_increase)
    previous_odds = row['odds_death']
    i += 1

death_age_cross_tab["odds_increase"] = odds_temp
```

print and plot odds increase

In [14]:
```
print(death_age_cross_tab["odds_increase"])
```
```
age_group
0 - 9 Years           NaN
10 - 19 Years    0.600253
20 - 29 Years    3.664247
30 - 39 Years    3.282448
40 - 49 Years    2.621245
50 - 59 Years    2.695426
60 - 69 Years    3.117961
70 - 79 Years    2.827796
80+ Years        2.633903
Name: odds_increase, dtype: float64
```

In [15]:
```
death_age_cross_tab["odds_increase"].plot(kind="bar").axhline(3)
```

Out[15]: <matplotlib.lines.Line2D at 0x1303ec3bd30>



for every 10 year increase in age, the odds of dying increase by approximately 3 times

---

create variable death_boolean = 1 if death_yn = "Yes", 0 otherwise

In [16]:
```
ds["death_boolean"] = np.where(ds["death_yn"]=='Yes',1,0)
```

In [17]:
```
print(ds["death_boolean"].value_counts())
```
```
0    5601811
1     158255
Name: death_boolean, dtype: int64
```

---

assess variable current_status

In [18]:
```
print(ds["current_status"].value_counts())
```
```
Laboratory-confirmed case    5462778
Probable Case                 297288
Name: current_status, dtype: int64
```

"Probable Case" sounds like "I'll marry you"

"Laboratory-confirmed case" sounds like a ring on the finger

will keep the two cohorts lumped together for now

Also, one field value follows book capitalization ("Probable Case")

and the other follows sentence capitalization ("Laboratory-confirmed case")

you go to obsessive-compulsive purgatory for that offense

---

death rates by age cohort

In [19]:
```
table_by_current_status = round(ds.pivot_table(["death_boolean"],
                                index=['age_group'],
                                columns=["current_status"],
                                margins=True)*100,2)
```

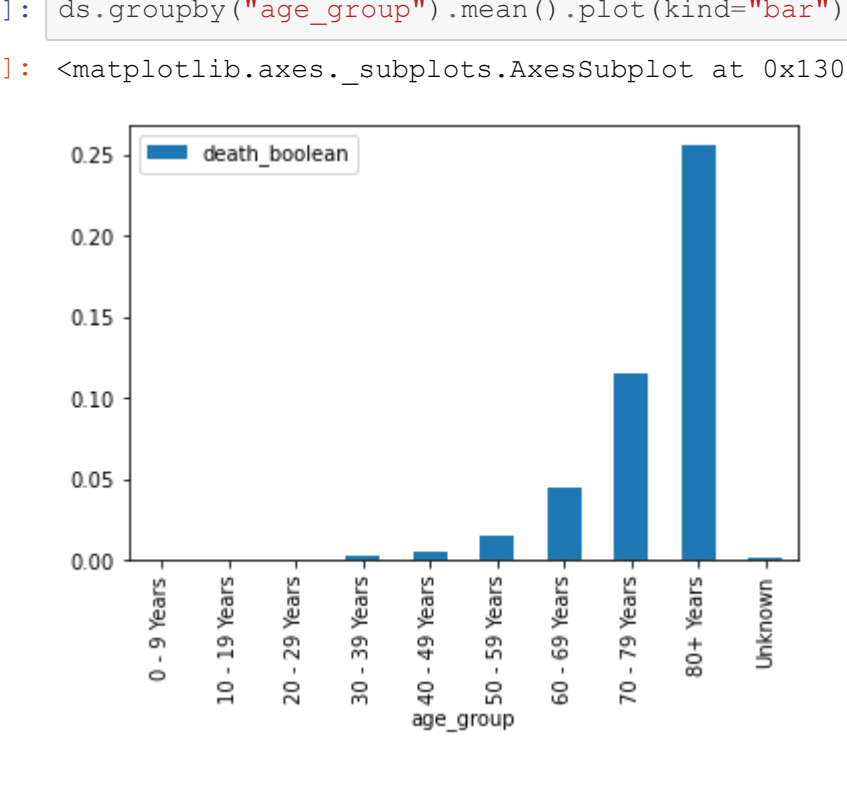
In [20]:
```
print(table_by_current_status.astype(str) + '%')
```
```
                           death_boolean
current_status Laboratory-confirmed case Probable Case     All
age_group
0 - 9 Years                        0.03%         0.02%   0.03%
10 - 19 Years                      0.02%          0.0%   0.02%
20 - 29 Years                      0.06%         0.08%   0.06%
30 - 39 Years                       0.2%         0.31%   0.21%
40 - 49 Years                      0.54%         0.72%   0.55%
50 - 59 Years                      1.44%         1.82%   1.46%
60 - 69 Years                      4.39%         4.89%   4.42%
70 - 79 Years                     11.43%        14.06%  11.55%
80+ Years                         24.87%        42.37%   25.6%
Unknown                            0.14%         0.18%   0.15%
All                                2.71%         3.42%   2.75%
```

---

plot mortality by age_group

In [21]:
```
ds.groupby("age_group").mean().plot(kind="bar")
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x130409e2460>

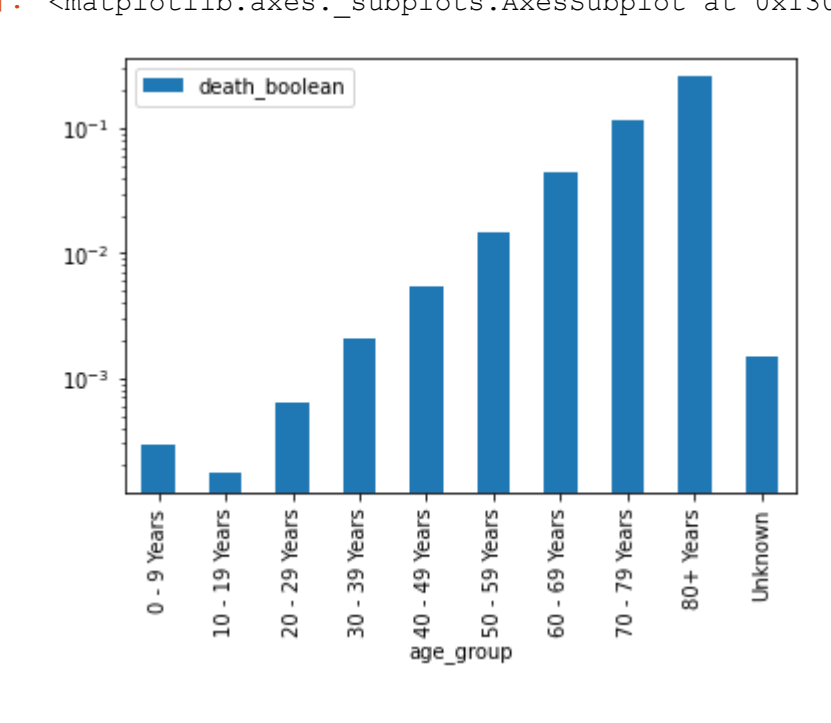


---

plot mortality by age_group with log scale

In [22]:
```
ds.groupby("age_group").mean().plot(kind="bar", log=True)
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1303f3fd640>



---

death counts by age_group

In [23]:
```
table_counts_by_current_status = ds.pivot_table(["death_boolean"],
                          index=['age_group'],
                          columns=["current_status"],
                          aggfunc=np.sum,
                          margins=True)
```

In [24]:
```
print(table_counts_by_current_status)
```
```
                           death_boolean
current_status Laboratory-confirmed case Probable Case     All
age_group
0 - 9 Years                           52             3      55
10 - 19 Years                         94             1      95
20 - 29 Years                        676            44     720
30 - 39 Years                       1848           128    1976
40 - 49 Years                       4482           316    4798
50 - 59 Years                      11565           844   12409
60 - 69 Years                      24715          1577   26292
70 - 79 Years                      36734          2209   38943
80+ Years                          67915          5031   72946
Unknown                                9             3      12
All                               148090         10156  158246
```

counts don't line up with aggregate results on CDC's website: 212,328

https://www.cdc.gov/nchs/nvss/vsrr/covid_weekly/index.htm, accessed 02 Nov 2020

---

assess variable medcond_yn

metadata defines variable as: "Presence of underlying comorbidity or disease"

In [25]:
```
print(ds["medcond_yn"].value_counts())
```
```
Missing    3976510
Yes         647375
Unknown     634512
No          501669
Name: medcond_yn, dtype: int64
```

In [26]:
```
print(pd.crosstab(ds["medcond_yn"], ds["death_yn"]))
```
```
death_yn    Missing       No  Unknown    Yes
medcond_yn
Missing     1956206  1576802   378134  65368
No            61685   413865    23244   2875
Unknown      157066   293021   162409  22016
Yes          146852   379267    53260  67996
```

comorbidities is a field on death certificates

so the dead population shouldn't have medcond_yn = "Unknown"

---

pivot table breaking out mortality by medcond_yn

In [27]:
```
table_by_medcond_yn = round(ds.pivot_table(["death_boolean"],
                          index=['age_group'],
                          columns=["medcond_yn"],
                          margins=True)*100,2)
```

In [28]:
```
print(table_by_medcond_yn.astype(str) + '%')
```
```
              death_boolean
medcond_yn          Missing      No Unknown     Yes     All
age_group
0 - 9 Years           0.02%   0.04%   0.01%   0.15%   0.03%
10 - 19 Years         0.01%   0.01%    0.0%   0.12%   0.02%
20 - 29 Years         0.04%   0.03%   0.09%   0.36%   0.06%
30 - 39 Years         0.13%   0.08%   0.23%   1.04%   0.21%
40 - 49 Years         0.34%   0.24%   0.63%   2.05%   0.55%
50 - 59 Years          0.9%   0.59%   1.47%    4.5%   1.46%
60 - 69 Years         2.86%   1.51%   4.16%  10.75%   4.42%
70 - 79 Years         7.82%   5.31%   11.8%  22.42%  11.55%
80+ Years            18.35%  20.23%   25.3%  42.54%   25.6%
Unknown               0.09%    0.0%   0.56%    0.0%   0.15%
All                   1.64%   0.57%   3.47%   10.5%   2.75%
```

Create pivot table to assess counts of deaths by medical condition status (variable medcond_yn)

In [29]:
```
pivot_death_by_medcond = ds.pivot_table(["death_boolean"],
                                        index=['age_group'],
                                        columns=["medcond_yn"],
                                        aggfunc = np.sum,
                                        margins=True)
```

In [30]:
```
pivot_death_by_medcond.style.format("{:,.0f}")
```

Out[30]:

	death_boolean				
medcond_yn age_group	Missing	No	Unknown	Yes	All
0 - 9 Years	31	10	2	12	55
10 - 19 Years	53	8	2	32	95
20 - 29 Years	349	41	95	235	720
30 - 39 Years	899	78	246	753	1,976
40 - 49 Years	2,106	181	621	1,890	4,798
50 - 59 Years	5,161	358	1,431	5,459	12,409
60 - 69 Years	10,974	473	2,878	11,967	26,292
70 - 79 Years	16,324	596	4,718	17,305	38,943
80+ Years	29,463	1,128	12,017	30,338	72,946
Unknown	6	0	6	0	12
All	65,366	2,873	22,016	67,991	158,246

Recode and reassess variable medcond_yn

In [31]:
```
ds["medcond_yn_recode"] = np.where(ds["medcond_yn"]=='Yes','Yes','No')
```

In [32]:
```
pivot_table_by_medcond_yn_recode = ds.pivot_table(["death_boolean"],
                                                  index=['age_group'],
                                                  columns=["medcond_yn_recode"],
                                                  aggfunc = np.sum,
                                                  margins=True)
```

In [33]:
```
pivot_table_by_medcond_yn_recode.style.format("{:,.0f}")
```

Out[33]:

	death_boolean		
medcond_yn_recode age_group	No	Yes	All
0 - 9 Years	43	12	55
10 - 19 Years	63	32	95
20 - 29 Years	485	235	720
30 - 39 Years	1,223	753	1,976
40 - 49 Years	2,908	1,890	4,798
50 - 59 Years	6,950	5,459	12,409
60 - 69 Years	14,325	11,967	26,292
70 - 79 Years	21,638	17,305	38,943
80+ Years	42,608	30,338	72,946
Unknown	12	0	12
All	90,255	67,991	158,246

assess mortality by medical condition

In [34]:
```
medcond_yes = ds[ds["medcond_yn_recode"]=="Yes"].groupby("age_group").mean()
```

In [35]:
```
medcond_no = ds[ds["medcond_yn_recode"]=="No"].groupby("age_group").mean()
```

In [36]:
```
medcond_combined = pd.merge(medcond_yes, medcond_no, how="inner", on="age_group")
```

In [37]:
```
print(round(medcond_combined * 100,2).astype(str) + "%")
```
```
              death_boolean_x death_boolean_y
age_group
0 - 9 Years             0.15%           0.02%
10 - 19 Years           0.12%           0.01%
20 - 29 Years           0.36%           0.05%
30 - 39 Years           1.04%           0.14%
40 - 49 Years           2.05%           0.37%
50 - 59 Years            4.5%           0.95%
60 - 69 Years          10.75%           2.96%
70 - 79 Years          22.42%           8.33%
80+ Years              42.54%          19.94%
Unknown                  0.0%           0.15%
```

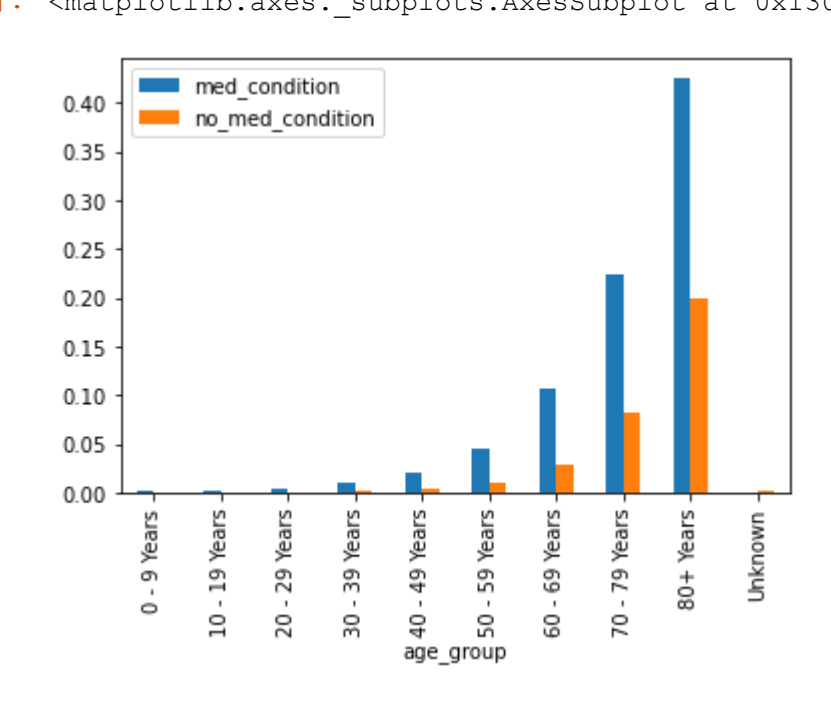
rename columns in dataset with mortality by medical condition and reassess

In [38]:
```
medcond_combined = medcond_combined.rename(columns = {"death_boolean_x": "med_condition",
                                "death_boolean_y": "no_med_condition"})
```

In [39]:
```
print(round(medcond_combined*100,2).astype(str) + "%")
```
```
              med_condition no_med_condition
age_group
0 - 9 Years           0.15%            0.02%
10 - 19 Years         0.12%            0.01%
20 - 29 Years         0.36%            0.05%
30 - 39 Years         1.04%            0.14%
40 - 49 Years         2.05%            0.37%
50 - 59 Years          4.5%            0.95%
60 - 69 Years        10.75%            2.96%
70 - 79 Years        22.42%            8.33%
80+ Years            42.54%           19.94%
Unknown                0.0%            0.15%
```

In [40]:
```
medcond_combined.plot(kind="bar")
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1303f5a8400>



create field for ratio of probabilities, medical condition vs. no medical condition

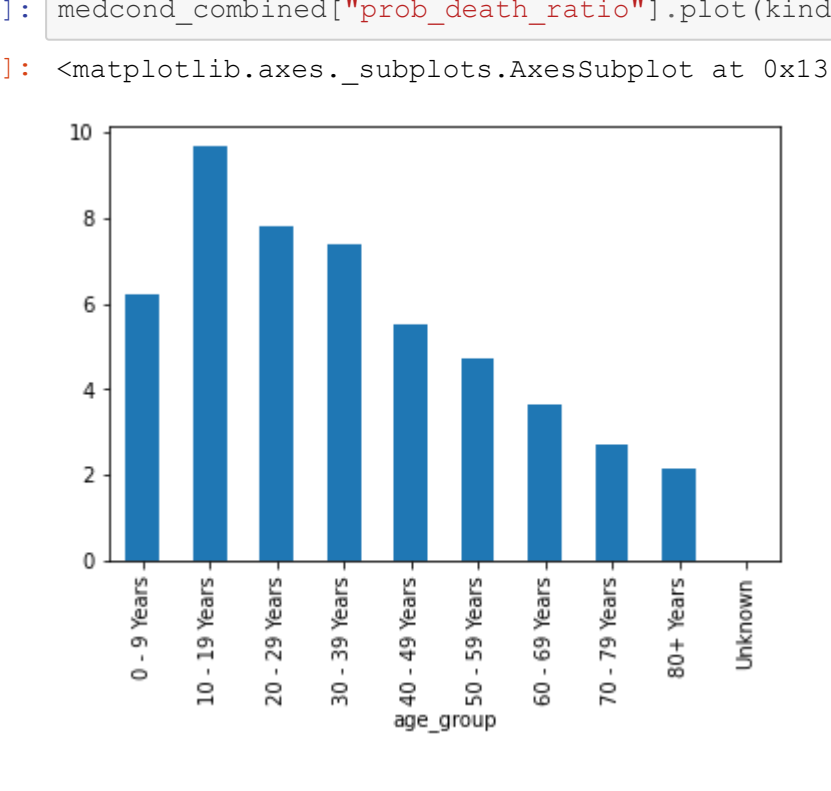
In [41]:
```
medcond_combined["prob_death_ratio"] = medcond_combined["med_condition"]/medcond_combined["no_med_condi
tion"]
```

In [42]:
```
medcond_combined["prob_death_ratio"]
```

Out[42]:
```
age_group
0 - 9 Years      6.216898
10 - 19 Years    9.679504
20 - 29 Years    7.817909
30 - 39 Years    7.378096
40 - 49 Years    5.529424
50 - 59 Years    4.724019
60 - 69 Years    3.633154
70 - 79 Years    2.693459
80+ Years        2.133212
Unknown          0.000000
Name: prob_death_ratio, dtype: float64
```

In [43]:
```
medcond_combined["prob_death_ratio"].plot(kind="bar")
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x130425f7070>



in the youngest age_group, those with pre-existing condition are ~10x more likely to die

but in 80+ year-old cohort, those with pre-existing conditions are "only" ~2x more likely to die

---

assess variable cdc_report_dt, which metadata defines as: "Initial case report date to CDC"

In [44]:
```
ds["cdc_report_dt"].isnull().value_counts()
```

Out[44]:
```
False    5760066
Name: cdc_report_dt, dtype: int64
```

the field is 100% populated, so that's what we'll go with for now

find range of dates and their frequency in dataset

In [45]:
```
ds["cdc_report_dt"].max()
```

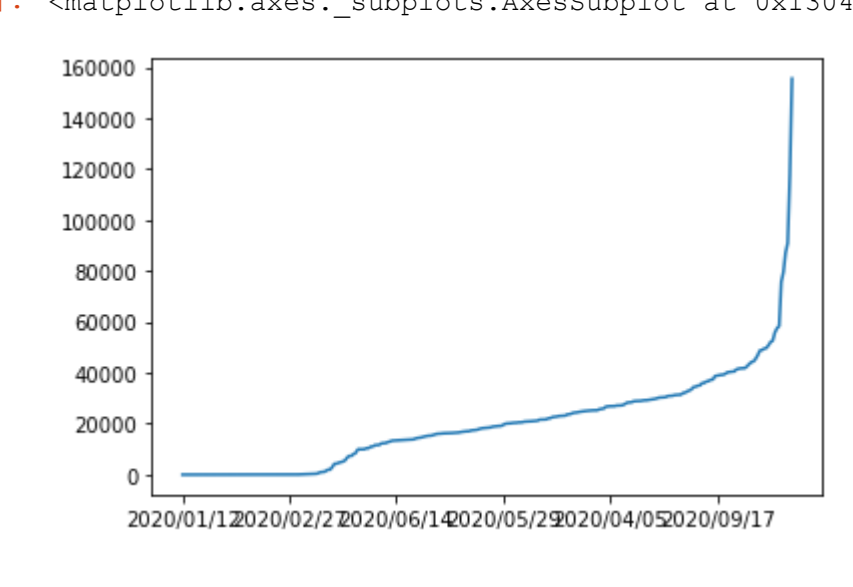
Out[45]: '2020/10/16'

the last date recorded is Oct 16, 2020, data is reasonably recent

plot frequency of dates

In [46]:
```
ds["cdc_report_dt"].value_counts(ascending=True).plot()
```

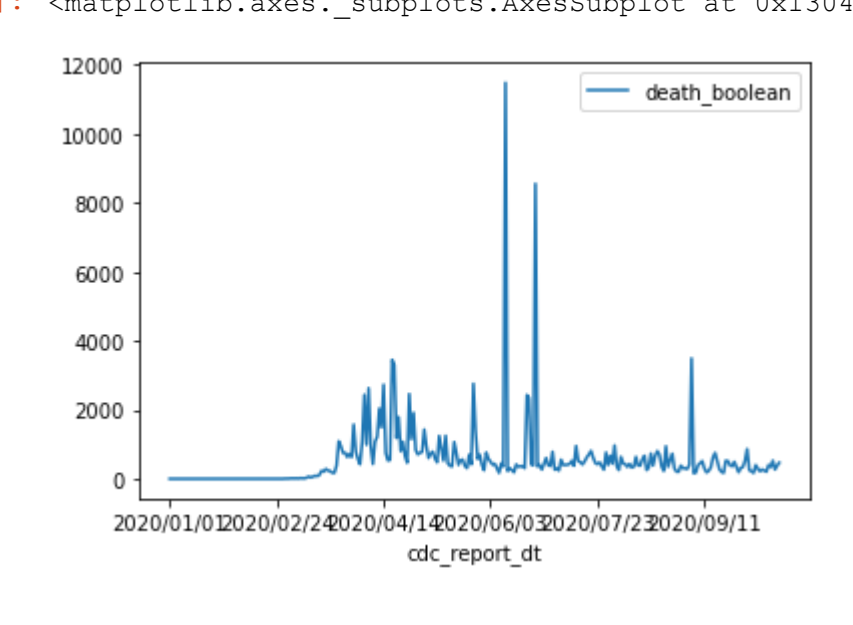
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x1304266e160>



plot frequency of deaths over time

In [47]:
```
ds[["cdc_report_dt","death_boolean"]].groupby("cdc_report_dt").sum().plot()
```

Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x130426cc880>



Those data spikes are odd, probably a reporting anomoly

In [48]:
```
ds[["cdc_report_dt","death_boolean"]].groupby("cdc_report_dt").sum().sort_values("death_boolean",ascend
ing=False)
```

Out[48]:

cdc_report_dt	death_boolean
2020/06/10	11480
2020/06/24	8552
2020/09/05	3493
2020/04/18	3449
2020/04/19	3299
...	...
2020/01/18	0
2020/01/17	0
2020/01/15	0
2020/01/14	0
2020/01/01	0

286 rows × 1 columns

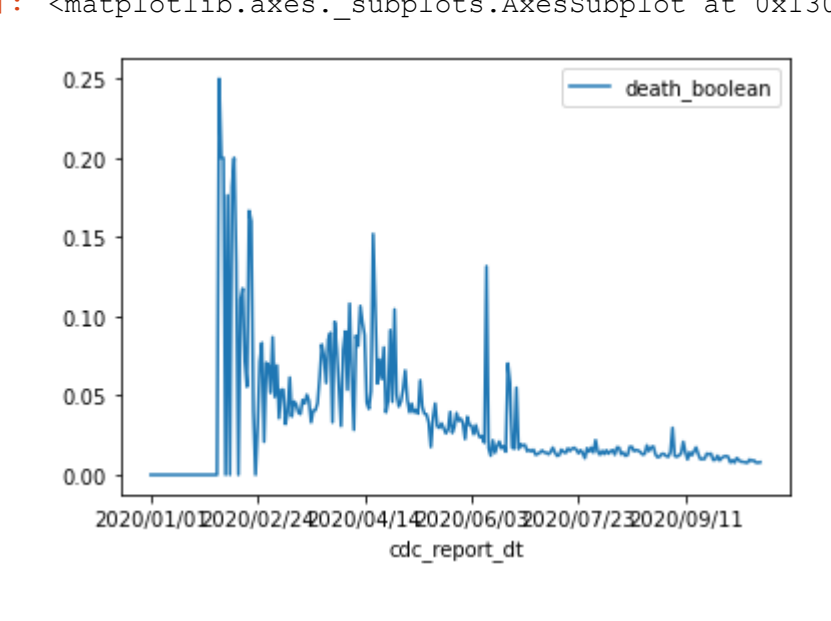
the largest two spikes dwarf the remaining spikes (3x to 4x larger)

---

assess mortality rates over time

In [49]:
```
ds[["cdc_report_dt","death_boolean"]].groupby("cdc_report_dt").mean().plot()
```

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x1304272c820>



the mortality rate seems to have setteled into a steady state

---

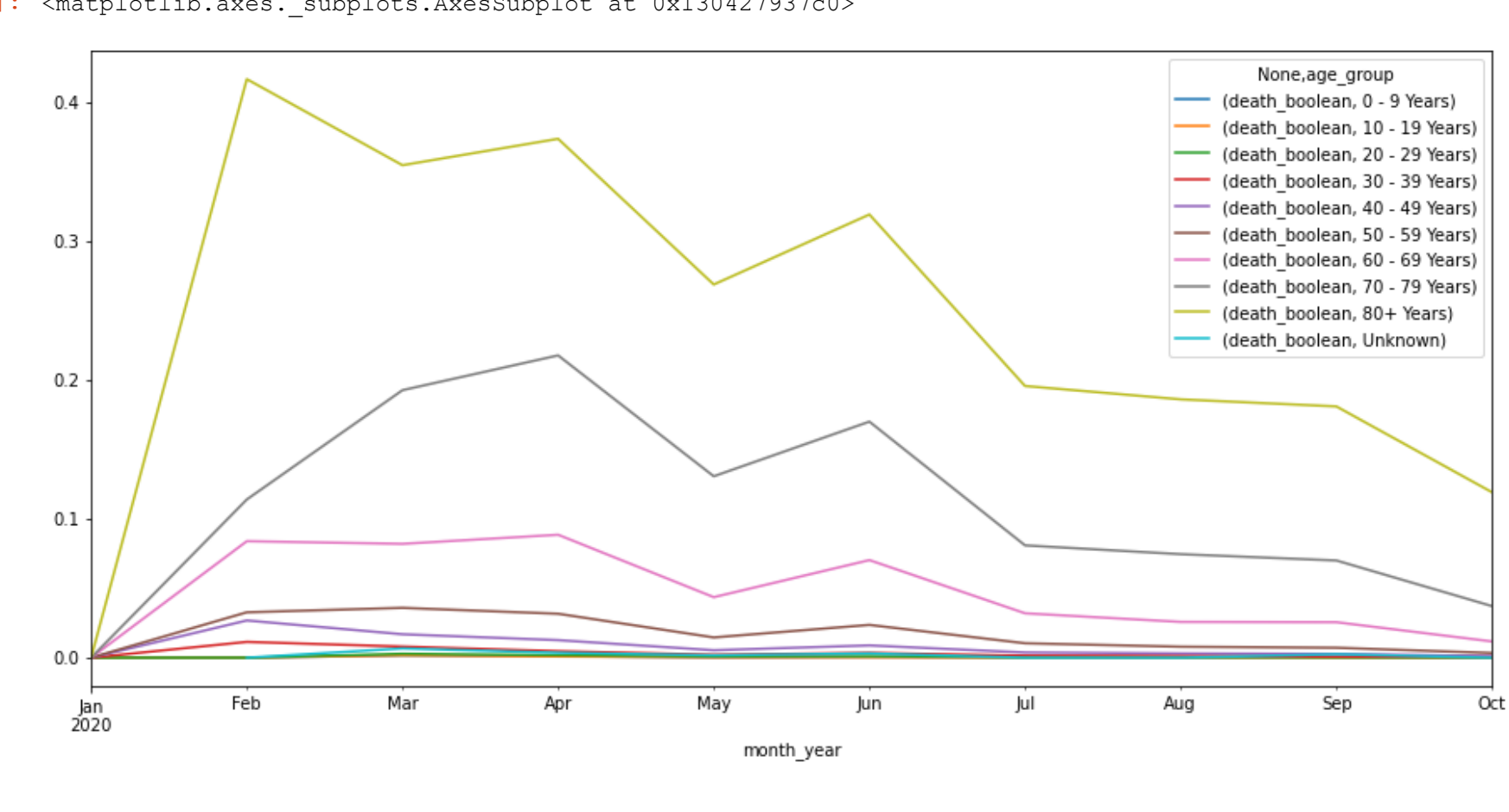
create month_year variable to smooth out noise with less granular aggregation

In [50]:
```
ds['month_year'] = pd.to_datetime(ds['cdc_report_dt']).dt.to_period('M')
```

plot mortality by age_group and month_year

In [51]:
```
ds.groupby(["month_year","age_group"]).mean().unstack().plot(figsize=(15,7))
```

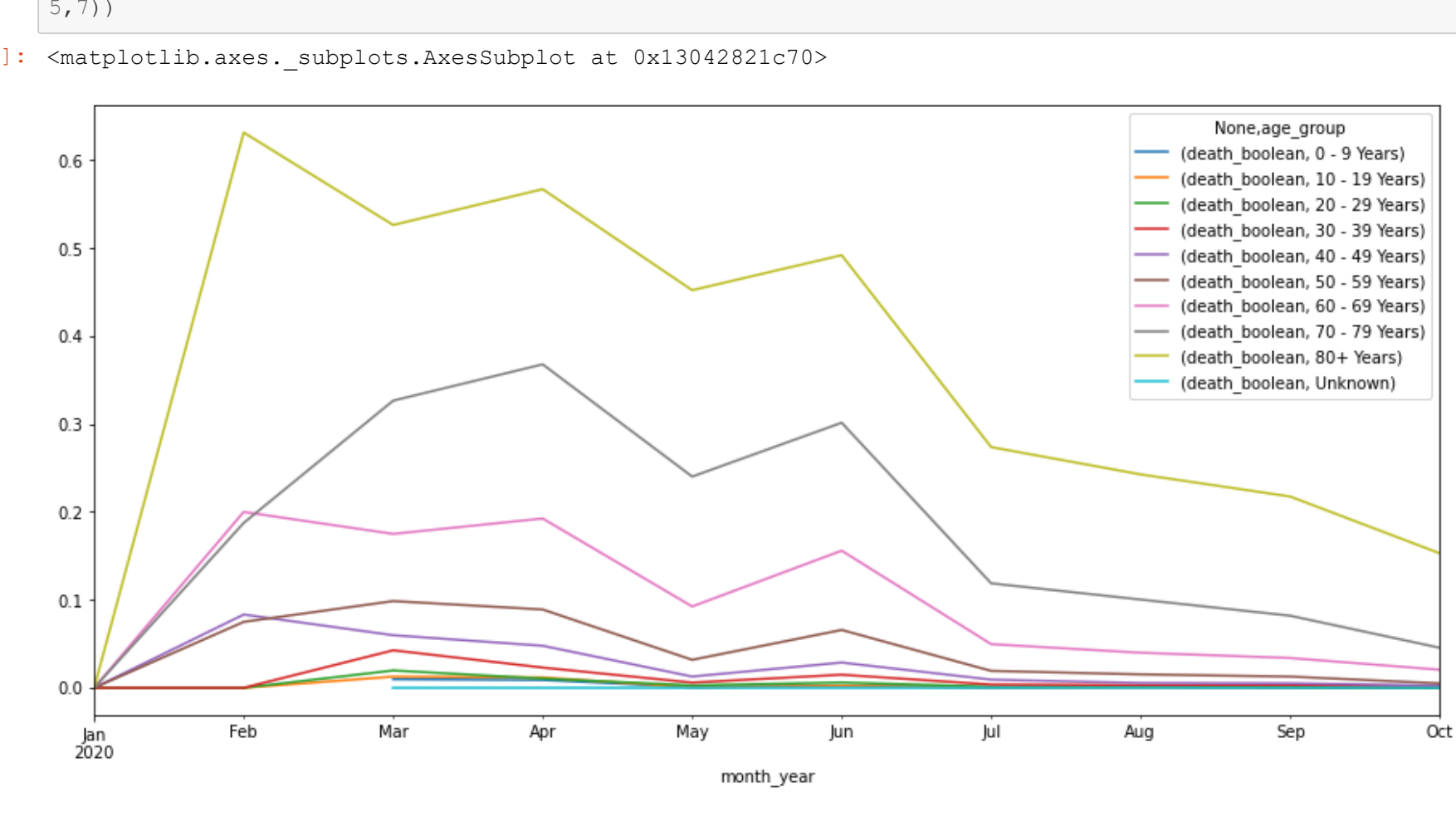
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x130427937c0>



plot mortality by age_group and month_year for medcond_yn_recode" = "Yes"

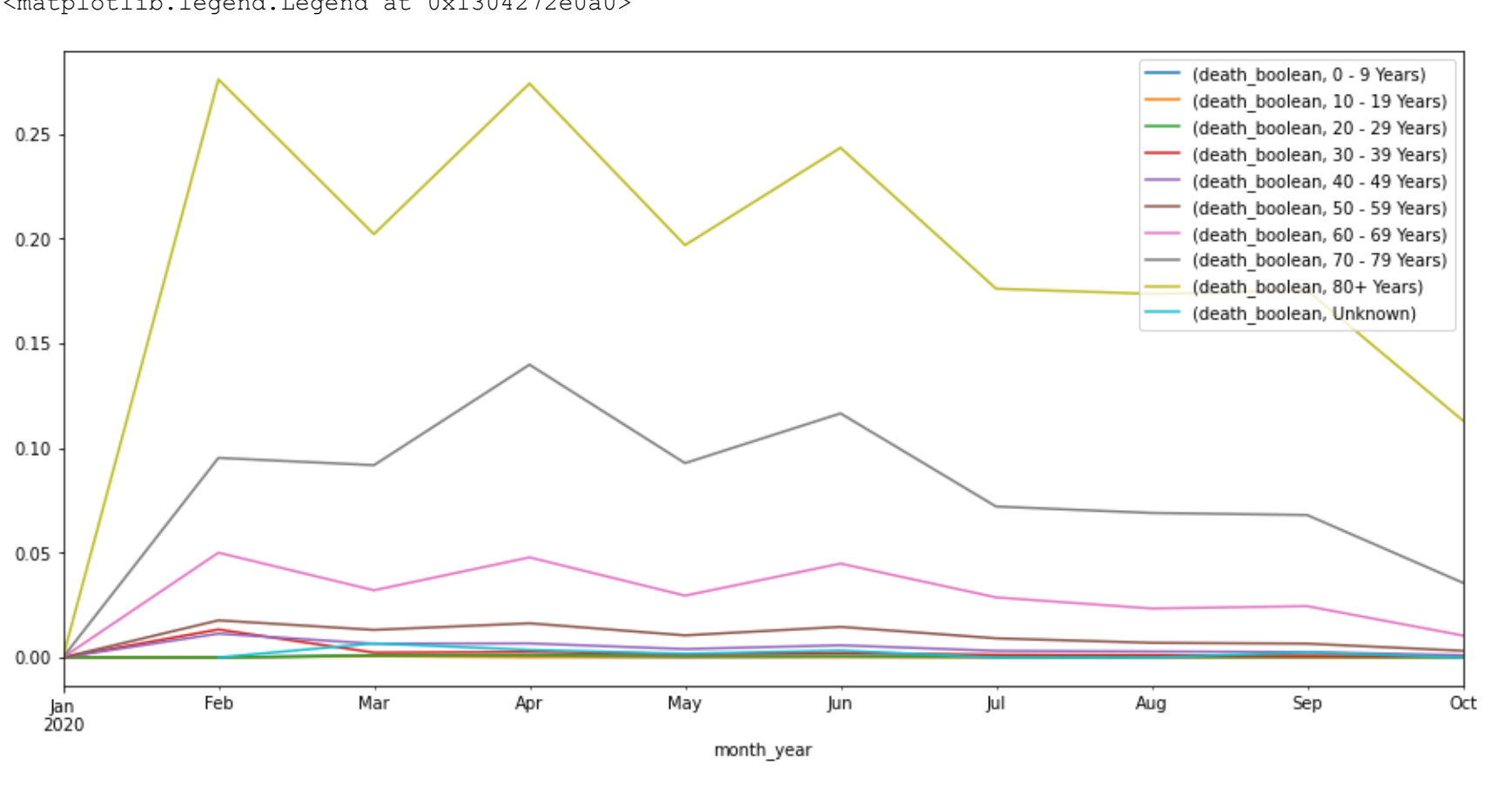
In [52]:
```
ds[ds["medcond_yn_recode"]=="Yes"].groupby(["month_year","age_group"]).mean().unstack().plot(figsize=(1
5,7))
```

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x13042821c70>



plot mortality by age_group and month_year for medcond_yn_recode" = "No"

In [53]:
```
ds[ds["medcond_yn_recode"]=="No"].groupby(["month_year","age_group"]).mean().unstack().plot(figsize=(15,7)).legend()
```
Out[53]: `<matplotlib.legend.Legend at 0x1304272e0a0>`



mortality is trending lower for all age_group cohorts

Download mortality dataset (2018), data available at:

https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm

Link to user guide (metadata) available at:

https://www.cdc.gov/nchs/data/dvs/Multiple_Cause_Record_Layout_2018-508.pdf

In [54]:
```
colspecs = [(19,20), (60,62), (62,63), (63,64), (64,66), (68,69), (69,73), (73,74), (74,76), (76,78), (78,80),
            (80,83), (83,84), (84,85), (101,105), (105,106), (106,107), (107,108), (108,109), (143,144), (144,145),
            (145,149), (149,152), (152,153), (153,156), (156,159), (159,161), (161,162), (162,164), (164,171),
            (171,178), (178,185), (185,192), (192,199), (199,206), (206,213), (213,220), (220,227), (227,234),
            (234,241), (241,248), (248,255), (255,262), (262,269), (269,276), (276,283), (283,290), (290,297),
            (297,304), (340,342), (343,348), (348,353), (353,358), (358,363), (363,368), (368,373), (373,378),
            (378,383), (383,388), (388,393), (393,398), (398,403), (403,408), (408,413), (413,418), (418,433),
            (423,428), (428,433), (433,438), (438,443), (443,444), (444,446), (446,447), (447,448), (448,449),
            (449,450), (483,486), (487,488), (488,490)]
```

In [55]:
```
mortality_url = "ftp://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/DVS/mortality/mort2018us.zip"
```

In [56]:
```
mortality_ds = pd.read_fwf(mortality_url, colspecs=colspecs, header=None)
```

Assign column names

In [57]:
```
mortality_ds.columns = [
"Resident_Status",
"Education_1989_revision",
"Education_2003_revision",
"Education_reporting_flag",
"Month_of_Death",
"Sex",
"Detail_Age",
"Age_Substitution_Flag",
"Age_Recode_52",
"Age_Recode_27",
"Age_Recode_12",
"Infant_Age_Recode_22",
"Marital_Status",
"Day_of_Week_of_Death",
"Current_Data_Year",
"Injury_at_Work",
"Manner_of_Death",
"Method_of_Disposition",
"Autopsy",
"Activity_Code",
"Place_of_Injury_for_Causes_W00Y34_except_Y06_and_Y07",
"ICDCode10thRevision",
"358_Cause_Recode",
"Reserved_Positions",
"113_Cause_Recode",
"130_Infant_Cause_Recode",
"39__Cause_Recode",
"ReservedPosition",
"Number_of_EntityAxis_Conditions",
"1stCondition_Entity",
"2ndCondition_Entity",
"3rdCondition_Entity",
"4thCondition_Entity",
"5thCondition_Entity",
"6thCondition_Entity",
"7thCondition_Entity",
"8thCondition_Entity",
"9thCondition_Entity",
"10thCondition_Entity",
"11thCondition_Entity",
"12thCondition_Entity",
"13thCondition_Entity",
"14thCondition_Entity",
"15thCondition_Entity",
"16thCondition_Entity",
"17thCondition_Entity",
"18thCondition_Entity",
"19thCondition_Entity",
"20thCondition_Entity",
"Number_of_RecordAxis_Conditions",
"1stCondition_Entity",
"2ndCondition_Entity",
"3rdCondition_Entity",
"4thCondition_Entity",
"5thCondition_Entity",
"6thCondition_Entity",
"7thCondition_Entity",
"8thCondition_Entity",
"9thCondition_Entity",
"10thCondition_Entity",
"11thCondition_Entity",
"12thCondition_Entity",
"13thCondition_Entity",
"14thCondition_Entity",
"15thCondition_Entity",
"16thCondition_Entity",
"17thCondition_Entity",
"18thCondition_Entity",
"19thCondition_Entity",
"20thCondition_Entity",
"Reserved_position",
"Bridged_Race",
"Bridged_Race_Flag",
"Race_Imputation_Flag",
"Allotherracesisimputed",
"Bridged",
"Hispanic_Origin",
"Hispanic_Origin_Bridged_Race_Recode",
"Race_Recode_40"]
```

create dummy variable for death

it's the mortality table, so the dummy variable doesn't add additional info--they're all dead

it just allows for easier summary statistic calculations for some functions

In [58]:
```
mortality_ds["death"]=1
```

create vector to subset mortality dataset by diseases of interest

In [59]:
```
conditions = mortality_ds["ICDCode10thRevision"].str[0:3].isin([

# https://icd.who.int/browse10/2019/en
# dataset says to use 2016 version, so hopefully they're backwards compatible
# below are all the codes for "Influenza and pneumonia (J09-J18)"

"J09", # Influenza due to identified zoonotic or pandemic influenza virus
"J10", # Influenza due to identified seasonal influenza virus
"J11", # Influenza, virus not identified
"J12", # Viral pneumonia, not elsewhere classified
"J13", # Pneumonia due to Streptococcus pneumoniae
"J14", # Pneumonia due to Haemophilus influenzae
"J15", # Bacterial pneumonia, not elsewhere classified
"J16", # Pneumonia due to other infectious organisms, not elsewhere classified
"J17", # Pneumonia in diseases classified elsewhere
"J18"  # Pneumonia, organism unspecified

])
```

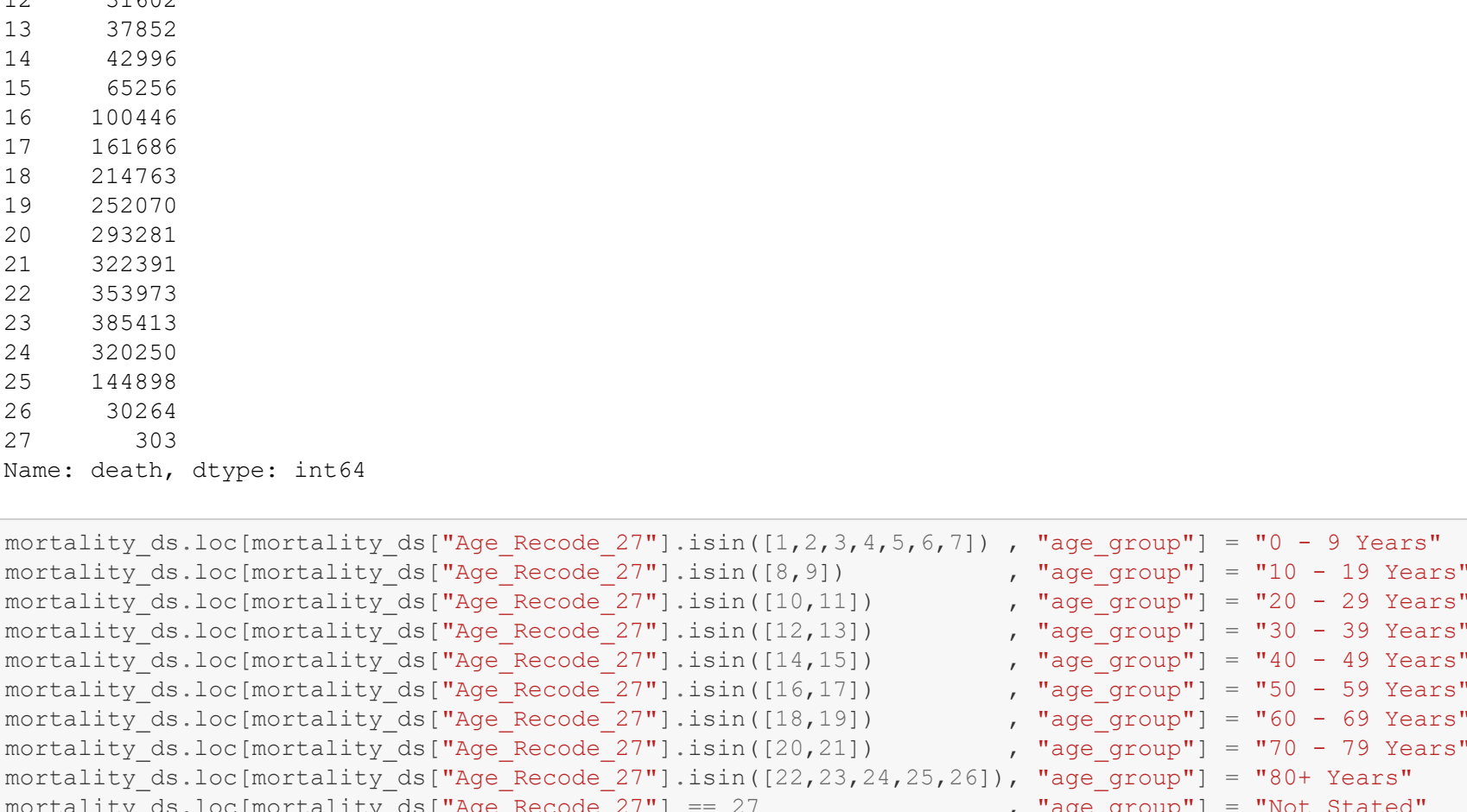
In [60]:
```
mortality_ds[conditions].sum()["death"]
```
Out[60]: 59279

59,279 people died from the above conditions according to one variable (field)

there are 40 other cause-of-death fields, 20 for "Record" cause of death

and 20 for "Entity" causes of death

In [61]:
```
mortality_ds[conditions].groupby(["Month_of_Death"]).sum()["death"].plot(kind="bar", figsize=(15,7))
```
Out[61]: `<matplotlib.axes._subplots.AxesSubplot at 0x1304284b520>`



In [71]:
```
mortality_ds.groupby(["Age_Recode_27"]).sum()["death"]
```
Out[71]:
```
Age_Recode_27
1      14367
2       7200
3       1455
4       1053
5        748
6        590
7       2357
8       3141
9      10430
10     19906
11     27614
12     31602
13     37852
14     42996
15     65256
16    100446
17    161686
18    214763
19    252070
20    293281
21    322391
22    353973
23    385413
24    320250
25    144898
26     30264
27       303
Name: death, dtype: int64
```

In [84]:
```
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([1,2,3,4,5,6,7]) , "age_group"] = "0 - 9 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([8,9])           , "age_group"] = "10 - 19 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([10,11])         , "age_group"] = "20 - 29 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([12,13])         , "age_group"] = "30 - 39 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([14,15])         , "age_group"] = "40 - 49 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([16,17])         , "age_group"] = "50 - 59 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([18,19])         , "age_group"] = "60 - 69 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([20,21])         , "age_group"] = "70 - 79 Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"].isin([22,23,24,25,26]), "age_group"] = "80+ Years"
mortality_ds.loc[mortality_ds["Age_Recode_27"] == 27                 , "age_group"] = "Not Stated"
```

In [85]:
```
mortality_ds.groupby(["age_group"]).sum()["death"]
```
Out[85]:
```
age_group
0 - 9 Years        27770
10 - 19 Years      13571
20 - 29 Years      47520
30 - 39 Years      69454
40 - 49 Years     108252
50 - 59 Years     262132
60 - 69 Years     466833
70 - 79 Years     615672
80+ Years        1234798
Not Stated           303
Name: death, dtype: int64
```

In [86]:
```
# https://www.census.gov/content/dam/Census/library/publications/2011/dec/c2010br-03.pdf
# 2010 was last census available.  Once 2020 data are published, it's possible to use a
# weighted average of the two to interpolate 2018 population

population_2010 = { "age_group" : [
    "0 - 9 Years",
    "10 - 19 Years",
    "20 - 29 Years",
    "30 - 39 Years",
    "40 - 49 Years",
    "50 - 59 Years",
    "60 - 69 Years",
    "70 - 79 Years",
    "80+ Years",
    ],

    "population" : [
    40550019,
    42717537,
    42687848,
    40141741,
    43599555,
    41962930,
    29253187,
    16595961,
    11236760
    ]
}
```

In [87]:
```
pd.DataFrame(population_2010, columns = ["age_group", "population"])
```
Out[87]:

	age_group	population
0	0 - 9 Years	40550019
1	10 - 19 Years	42717537
2	20 - 29 Years	42687848
3	30 - 39 Years	40141741
4	40 - 49 Years	43599555
5	50 - 59 Years	41962930
6	60 - 69 Years	29253187
7	70 - 79 Years	16595961
8	80+ Years	11236760

In [90]:
```
mortality_summary  = mortality_ds.groupby(["age_group"]).sum()["death"]
population_2010_ds = pd.DataFrame(population_2010, columns = ["age_group", "population"])
ds['Covid_Cases'] = 1
covid_summary = ds.groupby("age_group").sum()
```

In [107]:
```
intermediate_table = pd.merge(mortality_summary, population_2010_ds, how="inner", on="age_group")
final_summary_table = pd.merge(intermediate_table, covid_summary, how="inner", on="age_group")
```

In [111]:
```
final_summary_table = final_summary_table.rename(columns = {
                                        "death"        : "Total_Deaths",
                                        "population"   : "Population",
                                        "death_boolean": "Covid_Deaths"})
```

In [112]:
```
final_summary_table["Total_Mortality"] = final_summary_table["Total_Deaths"]/final_summary_table["Population"]
```

In [116]:
```
final_summary_table["Covid_Mortality"] = final_summary_table["Covid_Deaths"]/final_summary_table["Covid_Cases"]
```

In [127]:
```
final_summary_table["Mortality_Delta"] = \
    round((final_summary_table["Covid_Mortality"] - final_summary_table["Total_Mortality"]) * 100, 2).astype(str) + "%"
```

In [128]:
```
final_summary_table
```
Out[128]:

	age_group	Total_Deaths	Population	Covid_Deaths	Covid_Cases	Total_Mortality	Covid_Mortality	Mortality_Delta
0	0 - 9 Years	27770	40550019	55	189430	0.000685	0.000290	-0.04%
1	10 - 19 Years	13571	42717537	95	545036	0.000318	0.000174	-0.01%
2	20 - 29 Years	47520	42687848	720	1127849	0.001113	0.000638	-0.05%
3	30 - 39 Years	69454	40141741	1976	944365	0.001730	0.002092	0.04%
4	40 - 49 Years	108252	43599555	4798	877761	0.002483	0.005466	0.3%
5	50 - 59 Years	262132	41962930	12409	850025	0.006247	0.014598	0.84%
6	60 - 69 Years	466833	29253187	26292	595487	0.015958	0.044152	2.82%
7	70 - 79 Years	615672	16595961	38943	337082	0.037098	0.115530	7.84%
8	80+ Years	1234798	11236760	72946	284973	0.109889	0.255975	14.61%

In [ ]: