vaers

September 11, 2023

1 VAERS

1.1 https://github.com/jgarza9788/vaers

2 Disclaimer

THIS IS NOT MEDICAL ADVICE
if you want medical advice please go see your doctor.

please also read the disclaimer on the VAERS data on their website.
VAERS website

3 get started

3.1 1. get the data

download the data from VAERS website
link to VAERS website

make sure to download the 2020-2022 data 2022VAERSVAX.csv
2022VAERSSYMPTOMS.csv
2022VAERSDATA.csv
2021VAERSVAX.csv
2021VAERSSYMPTOMS.csv
2021VAERSDATA.csv
2020VAERSVAX.csv
2020VAERSSYMPTOMS.csv
2020VAERSDATA.csv

3.2 2. move and unzip

move the AllVAERSDataCSVS.zip to VAERS folder and unzip it

3.3 3. check the files

you should have at least 6 files

```
.../VAERS/AllVAERSDataCSVS/
    2022VAERSVAX.csv
    2022VAERSSYMPTOMS.csv
    2022VAERSDATA.csv
```

```
...
2020VAERSVAX.csv
2020VAERSSYMPTOMS.csv
2020VAERSDATA.csv
```

4 THE CODE!

4.1 setup/imports

[1]:

```python
# Import needed libraries
import pandas as pd
import numpy as np
import os, sys, re, json
from collections import Counter

# used for graphs/charts
# import plotly.express as px  # too large
import matplotlib.pyplot as plt
import seaborn as sns

from IPython.display import display, HTML # needed for IPYNB
```

[2]:

```python
# set pandas options
pd.options.display.float_format = '{:,.2f}'.format
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', None)
```

4.2 directory variables

DIR (String) : the current directory this file is in
DATAPATH (String) : the location where the data files are stored

[3]:

```python
# DIR = os.path.dirname(os.path.realpath(__file__)) #doesn't work due to IPYNB␣
 ↪file
DIR = '.' #this works in IPYNB
DATAPATH = os.path.join(DIR,'AllVAERSDataCSVS')

print('DATAPATH: ',DATAPATH)
```

```
DATAPATH:  .\AllVAERSDataCSVS
```

4.3 Saving and Loading JSON files

[4]:

```python
# LOADING AND SAVING JSON FILES

def load_json(path_to_file):
    """
    loads json from file
    """
    with open(path_to_file, 'r') as f:
        return json.load(f)

def save_json(path_to_file,data):
    """
    writes dict/json to file
    """
    with open(path_to_file, 'w') as f:
        json.dump(data, f,indent=4)

## test
# save_json(os.path.join(DATAPATH,'test.json'),{"HELLO":0})
# print(load_json(os.path.join(DATAPATH,'test.json')))
```

4.4 processing files

the functions flow will process the csv files into one all_data.json file

[5]:

```
def has_covid(text):
    """
    returns 1 or 0 if text has \'COVID\' in it
    """
    if re.search('COVID',text.upper()):
        return 1
    else:
        return 0

def compile_files(directory,files):
    """
    compiles/adds/unions multiple csv files together and returns a dataframe
    """
    df = pd.DataFrame()
    for i,f in enumerate(files):
        df0 = pd.read_csv(os.path.
 ↪join(directory,f),encoding='cp1252',low_memory=False)
        df = pd.concat([df,df0])
    return df

def process_to_one_file():
    """
    processes VAERS data from 2020 - 2022, creates all_data.json, and returns a␣
 ↪dataframe
    """
    print('process to one file\n\tthis might take a while...go get a drink    \n')
    df_data = compile_files(DATAPATH,['2020VAERSDATA.csv','2021VAERSDATA.csv','2022VAERSDATA.csv'])
    df_vax = compile_files(DATAPATH,['2020VAERSVAX.csv','2021VAERSVAX.csv','2022VAERSVAX.csv'])
    df_sym = compile_files(DATAPATH,['2020VAERSSYMPTOMS.csv','2021VAERSSYMPTOMS.csv','2022VAERSSYMPTOMS.csv'])

    # print("""
    # symptoms are contained in columns (up to 5 symptoms per event)
    # we must transform these symptoms into a single list for each event
    # """)
    print('dedup-ing Symptoms')
    vid = list(df_sym['VAERS_ID'].unique())

    symptom_columns = [
        'SYMPTOM1',
        'SYMPTOM2',
        'SYMPTOM3',
        'SYMPTOM4',
        'SYMPTOM5'
    ]

    idf_sym = []
    for index,v in enumerate(vid):
        if index%5000 == 0:
            print('{:.2f}'.format(index/len(vid)), end='\r')
        temp = df_sym[df_sym['VAERS_ID'] == v]
        temp = temp.to_dict(orient='records')
        syms = []
        for t in temp:
            for sc in symptom_columns:
                if isinstance(t[sc],str):
                    syms.append(t[sc])
        idf_sym.append({'VAERS_ID':v,'SYMPTOMS':syms})
    df_sym = pd.DataFrame(idf_sym)

    print('merge data')
    df  = pd.merge(df_data,df_vax,how='outer',on='VAERS_ID')
    df  = df.drop_duplicates(ignore_index = True)
    df  = pd.merge(df,df_sym,how='outer',on='VAERS_ID')

    df.reset_index()

    # creating a new column depending if this is the covid vaccine or not
    df['COVID_VAX'] = df['VAX_TYPE'].apply(has_covid)
    df = df[df['COVID_VAX'] == 1]
    # print(len(df))

    #save json file
    json_file = os.path.join(DATAPATH,'all_data.json')
    save_json(json_file,df.to_dict(orient='records'))
    print('saved: ',json_file)

    #save out csv file (not needed), but people might like a csv
    csv_file = os.path.join(DATAPATH,'all_data.csv')
    df.to_csv(csv_file)
    print('saved: ',csv_file)

    return df
```

4.5 get data

the below will get data from the files or from all_data.json.

> please note i am only using 25% of the data below, but you can see the .pdf for how this will look with all the data, or edit and run the notebook on your own.

df (DataFrame) : contains all the datafrom VAERS files listed below * .../VAERS/AllVAERSDataCSVS/
* 2022VAERSVAX.csv
* 2022VAERSSYMPTOMS.csv
* 2022VAERSDATA.csv
* ... * 2020VAERSVAX.csv
* 2020VAERSSYMPTOMS.csv
* 2020VAERSDATA.csv

[6]:

```
def get_data():
    """
    gets the data and returns a dataframe
    """

    all_data = os.path.join(DATAPATH,'all_data.json')

    if os.path.isfile(all_data):
        print('loading all_data.json (15sec-30sec)')
        df = load_json(all_data)
        df = pd.DataFrame(df)
    else:
        print('processing the 2020-202X files')
        print("""
        .../VAERS/AllVAERSDataCSVS/
            20??VAERSVAX.csv
            20??VAERSSYMPTOMS.csv
            20??VAERSDATA.csv
        """)
        df = process_to_one_file()

    return df

df = get_data() # this will get all the data
# df = get_data().sample(int(946527 * 0.25)) # only 25% of the data

# assuming null for these is No
df.loc[df['DIED'].isna(),'DIED'] = 'N'
df.loc[df['L_THREAT'].isna(),'L_THREAT'] = 'N'
df.loc[df['ER_VISIT'].isna(),'ER_VISIT'] = 'N'
df.loc[df['HOSPITAL'].isna(),'HOSPITAL'] = 'N'

print('\nloaded {:,} records/rows\n'.format(len(df)))
print('columns:\n',df.columns.to_list())
print('\n\ndf.head(10):\n')
display(df.head(2))
```

```
processing the 2020-202X files

        …/VAERS/AllVAERSDataCSVS/
            20??VAERSVAX.csv
            20??VAERSSYMPTOMS.csv
            20??VAERSDATA.csv

process to one file
        this might take a while…go get a drink

dedup-ing Symptoms
merge data
saved:  .\AllVAERSDataCSVS\all_data.json
saved:  .\AllVAERSDataCSVS\all_data.csv

loaded 983,305 records/rows

columns:
 ['VAERS_ID', 'RECVDATE', 'STATE', 'AGE_YRS', 'CAGE_YR', 'CAGE_MO', 'SEX',
'RPT_DATE', 'SYMPTOM_TEXT', 'DIED', 'DATEDIED', 'L_THREAT', 'ER_VISIT',
'HOSPITAL', 'HOSPDAYS', 'X_STAY', 'DISABLE', 'RECOVD', 'VAX_DATE', 'ONSET_DATE',
'NUMDAYS', 'LAB_DATA', 'V_ADMINBY', 'V_FUNDBY', 'OTHER_MEDS', 'CUR_ILL',
'HISTORY', 'PRIOR_VAX', 'SPLTTYPE', 'FORM_VERS', 'TODAYS_DATE', 'BIRTH_DEFECT',
'OFC_VISIT', 'ER_ED_VISIT', 'ALLERGIES', 'VAX_TYPE', 'VAX_MANU', 'VAX_LOT',
'VAX_DOSE_SERIES', 'VAX_ROUTE', 'VAX_SITE', 'VAX_NAME', 'SYMPTOMS', 'COVID_VAX']
```

df.head(10):

```
       VAERS_ID    RECVDATE STATE  AGE_YRS  CAGE_YR  CAGE_MO SEX RPT_DATE  \
46308    902418  12/15/2020    NJ    56.00    56.00      NaN   F      NaN   
46327    902440  12/15/2020    AZ    35.00    35.00      NaN   F      NaN   

                                                                                ␣
 ↪                         SYMPTOM_TEXT  \
46308  Patient experienced mild numbness traveling from injection site up and␣
 ↪down arm that subsided over 20 minutes.
46327                                                                           ␣
 ↪                         C/O Headache

      DIED DATEDIED L_THREAT ER_VISIT HOSPITAL  HOSPDAYS X_STAY DISABLE  \
46308    N       NaN        N        N        N       NaN    NaN     NaN   
46327    N       NaN        N        N        N       NaN    NaN     NaN   

      RECOVD    VAX_DATE  ONSET_DATE  NUMDAYS LAB_DATA V_ADMINBY V_FUNDBY  \
46308      Y  12/15/2020  12/15/2020     0.00     none       PVT      NaN   
46327      Y  12/15/2020  12/15/2020     0.00     none       PVT      NaN   

      OTHER_MEDS CUR_ILL HISTORY PRIOR_VAX SPLTTYPE  FORM_VERS TODAYS_DATE  \
46308      latex    none    none       NaN      NaN          2  12/15/2020   
46327        NaN     NaN     NaN       NaN      NaN          2  12/15/2020   

      BIRTH_DEFECT OFC_VISIT ER_ED_VISIT ALLERGIES VAX_TYPE         VAX_MANU  \
46308          NaN       NaN         NaN      none  COVID19  PFIZER\BIONTECH   
46327          NaN       NaN         NaN       NaN  COVID19  PFIZER\BIONTECH   

       VAX_LOT VAX_DOSE_SERIES VAX_ROUTE VAX_SITE  \
46308   EH9899               1        IM       LA   
46327  EH 9899               1       SYR       LA   

                                  VAX_NAME  \
46308  COVID19 (COVID19 (PFIZER-BIONTECH))   
46327  COVID19 (COVID19 (PFIZER-BIONTECH))   

                                            SYMPTOMS  COVID_VAX  
46308  [Hypoaesthesia, Injection site hypoaesthesia]          1  
46327                                     [Headache]          1  
```

[7]:

```python
def print_row(items,column_lengths=[]):
    """
    comes in handly :)
    """
    row = ''
    for index,i in enumerate(items):
        try:
            cl = column_lengths[index]
        except IndexError:
            cl = 20
        row += str(i)[0:cl].ljust(cl)
    print(row)
```

4.6 gets a list of symptoms

df_symptoms (DataFrame) : a list of all the symtoms and the counts of each all_symptoms (list): a list of all the symtoms and the counts of each

note:
symptoms might be medical jargon or plain english
i.e. "RASH","ERYTHEMA", and "ITCHY RED SKIN"
would be reported as different items (for now)

note:
the counts/percentages below are of the symptoms.
and one adverse reaction can have multiple symptoms.

[33]:

```
def get_symptom_list(df,column='SYMPTOMS'):
    """
    returns a list of symptoms for the dataframe
    """
    s = df[column].to_list()
    l = []
    for i in s:
        try:
            for j in i:
                if str(j) == 'nan':
                    pass
                else:
                    l.append(str(j).upper())
        except:
            pass
    return l

all_symptoms =  get_symptom_list(df,'SYMPTOMS')
symptoms_count = len(all_symptoms)
all_symptoms = Counter(all_symptoms).most_common()

df_symptoms = pd.DataFrame(all_symptoms,columns=['SYMPTOM','COUNT'])
df_symptoms['PERCENT'] = (df_symptoms['COUNT']/symptoms_count)*100

file_name = os.path.join(DATAPATH,'symptoms.csv')
df_symptoms.to_csv(file_name)
print('saved: ',file_name)
```

```
topX = 50
print('below are the top {} symptoms'.format(topX))
display(df_symptoms.head(topX))
```

```
saved:  .\AllVAERSDataCSVS\symptoms.csv
below are the top 50 symptoms
                         SYMPTOM   COUNT  PERCENT
0                       HEADACHE  154222     3.64
1                        PYREXIA  133598     3.15
2                        FATIGUE  133167     3.14
3                           PAIN  110322     2.60
4                         CHILLS  108458     2.56
5                       COVID-19   92983     2.19
6                         NAUSEA   84493     1.99
7              PAIN IN EXTREMITY   82660     1.95
8                      DIZZINESS   81874     1.93
9       SARS-COV-2 TEST POSITIVE   65186     1.54
10                       MYALGIA   59229     1.40
11                      DYSPNOEA   56400     1.33
12                    ARTHRALGIA   53648     1.26
13           INJECTION SITE PAIN   46943     1.11
14                          RASH   44233     1.04
15                      PRURITUS   39742     0.94
16                      ASTHENIA   38566     0.91
17                      VOMITING   37968     0.89
18                         COUGH   37029     0.87
19  EXPIRED PRODUCT ADMINISTERED   32188     0.76
20                     DIARRHOEA   30722     0.72
21               SARS-COV-2 TEST   30267     0.71
22       INJECTION SITE ERYTHEMA   30030     0.71
23              NO ADVERSE EVENT   29044     0.68
24              FEELING ABNORMAL   29000     0.68
25                       MALAISE   28990     0.68
26         PRODUCT STORAGE ERROR   28557     0.67
27                    CHEST PAIN   28460     0.67
28                     URTICARIA   26870     0.63
29                  PARAESTHESIA   26726     0.63
30               LYMPHADENOPATHY   26171     0.62
31                      ERYTHEMA   26053     0.61
32                 HYPOAESTHESIA   25670     0.61
33       INJECTION SITE SWELLING   25562     0.60
34                 HYPERHIDROSIS   25552     0.60
35           PERIPHERAL SWELLING   21227     0.50
36                    BLOOD TEST   20948     0.49
37          CONDITION AGGRAVATED   20896     0.49
38       INJECTION SITE PRURITUS   20252     0.48
```

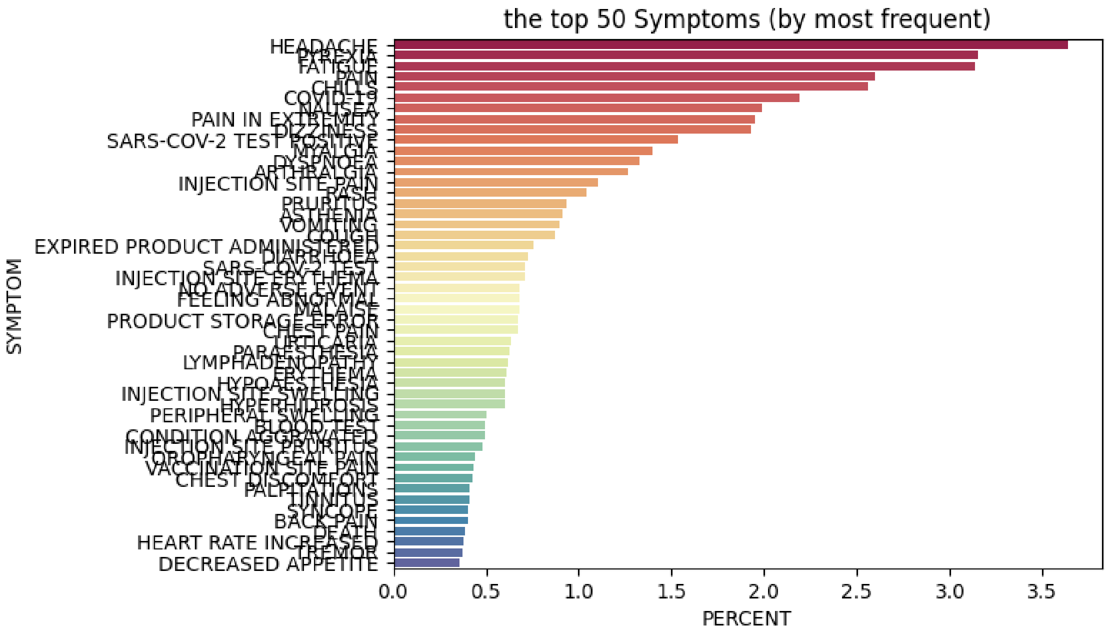
```
39           OROPHARYNGEAL PAIN   18683     0.44
40       VACCINATION SITE PAIN   18342     0.43
41            CHEST DISCOMFORT   18170     0.43
42                PALPITATIONS   17611     0.42
43                    TINNITUS   17336     0.41
44                     SYNCOPE   17070     0.40
45                   BACK PAIN   16984     0.40
46                       DEATH   16320     0.38
47        HEART RATE INCREASED   16234     0.38
48                      TREMOR   15864     0.37
49          DECREASED APPETITE   15183     0.36
```

[35]:

```python
# lets put it in a histogram!

# fig = px.histogram(
#     df_symptoms.head(topX),
#     x="SYMPTOM",
#     y="PERCENT",
#     # marginal="rug", # can be `box`, `violin`
#      hover_data=df_symptoms.head(topX).columns
#     )
# fig.show()

sns.barplot(
    data=df_symptoms.head(topX),
    x="PERCENT",
    y="SYMPTOM",
    # hue="PERCENT"
    palette = "Spectral"
    ).set(title=f'the top {topX} Symptoms (by most frequent)')
```

[35]:

```
[Text(0.5, 1.0, 'the top 50 Symptoms (by most frequent)')]
```



4.7 break down functions...

these functions will help me breakdown the data

4.7.1 break_down_columns

```
[36]: def break_down_columns(idf,column):
          """
          shows what values there are for a given column (with counts and percent)
          """
          print('\nbreak down of {0}'.format(column))
          idf = pd.DataFrame(idf[column])
          idf = idf.fillna('nan')

          result = []

          l = list(idf[column].unique())
          for i in l:
              df0 = idf[idf[column]==i]
              result.append({'column':column,'value':i,'count':len(df0),'percent':
       ↪(len(df0)/len(idf))*100})

          result = pd.DataFrame(result)
          display(result)
          return result
```

```
# test
# break_down_columns(df,'DIED')
```

4.7.2 break_down_buckets

[37]:

```
def break_down_buckets(idf,column,buckets,message='',nan_value=-1):
    """
    breaks a column down into buckets/bins
    """
    idf = idf.fillna(nan_value)

    print('',message,'\ncolumn: ',column, '\nbuckets: ', buckets)

    idf = pd.DataFrame(idf[column])
    idf['bucket'] = pd.cut(idf[column], bins=buckets)
    idf = idf.groupby(by='bucket').count()
    idf['percent'] = (idf[column]/idf[column].sum())*100
    idf['percent'] = idf['percent'].round(2)
    display(idf)

    return idf

#test
# break_down_buckets(df,'AGE_YRS',[-1,0,15,25,35,45,55,65,75,85,500])
```

4.8 Additional Numbers

~~these numbers are **not** part of the vaers data,~~
~~however they are important to analyzing the data~~

~~vaxx (int) : the number of vaccinated (1 or more shots) US citizens according to a quick google search (on 8/3/2021)~~

~~**google no longer shows partially vaxxed** so we are using the **Total doses given** number "Total doses given" shows the number of vaccine doses given to people. Since some vaccines require more than 1 dose, the number of fully vaccinated people is likely lower. "People fully vaccinated" shows how many people have received the full amount of doses for the COVID-19 vaccine. ~~

~~This number was updated on 06/28/2022~~

~~full_vaxx (int) : the number of vaccinated (2 or more shots) US citizens according to a quick google search (on 06/28/2022)~~

~~the numbers were taken from https://usafacts.org/visualizations/covid-vaccine-tracker-states on 11/18/2022~~

the numbers were taken from https://usafacts.org/visualizations/covid-vaccine-tracker-states on 9/10/2023

[79]:
```
vaxx = 270_227_181 + 230_637_348
full_vaxx = 230_637_348
```

[80]:
```
#only death vaers
df_death = df[df['DIED']=='Y']
```

4.9 Analyzing the df (all_data.json)

this is just a generic break down of the data

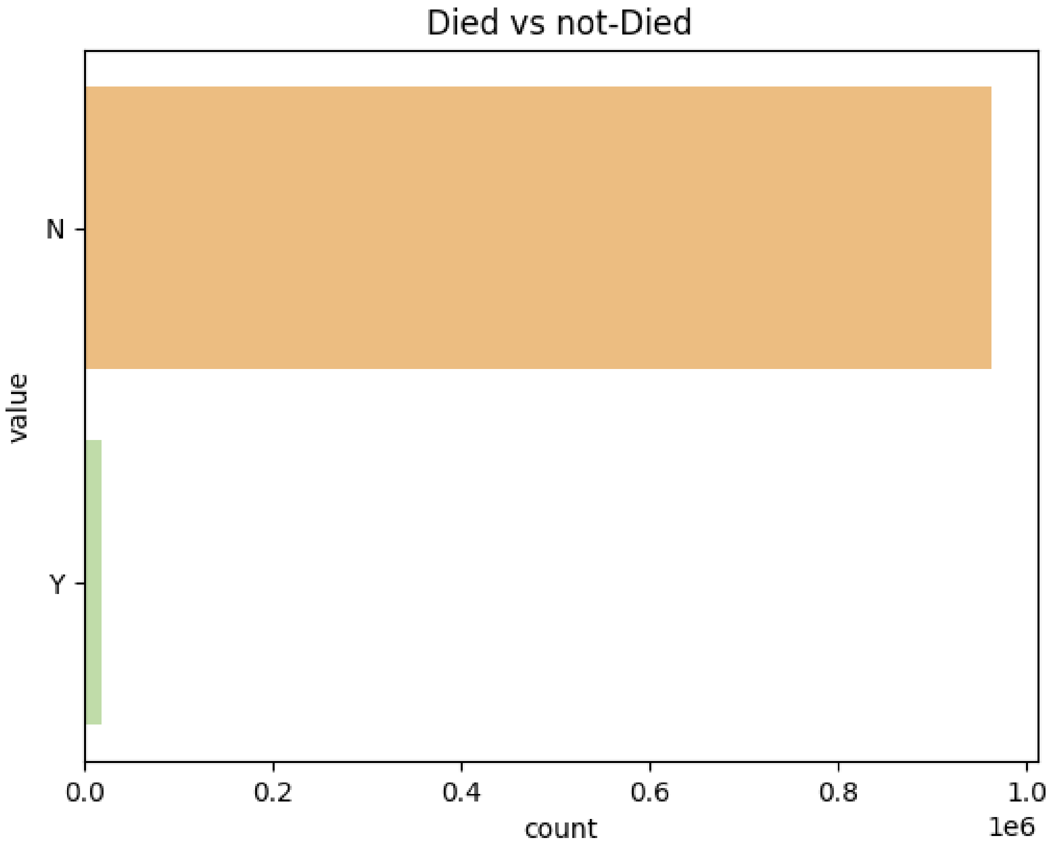
df_death (DataFrame) : adverse reactions that resulted in a death

4.9.1 DIED column

[81]:
```
temp = break_down_columns(df,'DIED')
print('\n' + '-'*25 + '\n')



# fig = px.pie(
#     temp,
#     values='count',
#     names='value',
#     title='Death as Symptoms from Vaccine',
#     # hover_data = temp.columns
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="count",
    y="value",
    # hue="PERCENT"
    palette = "Spectral"
    ).set(title='Died vs not-Died')
```

```
break down of DIED

  column value   count  percent
0   DIED     N  963477    97.98
1   DIED     Y   19828     2.02


-------------------------
```

[81]:
```
[Text(0.5, 1.0, 'Died vs not-Died')]
```

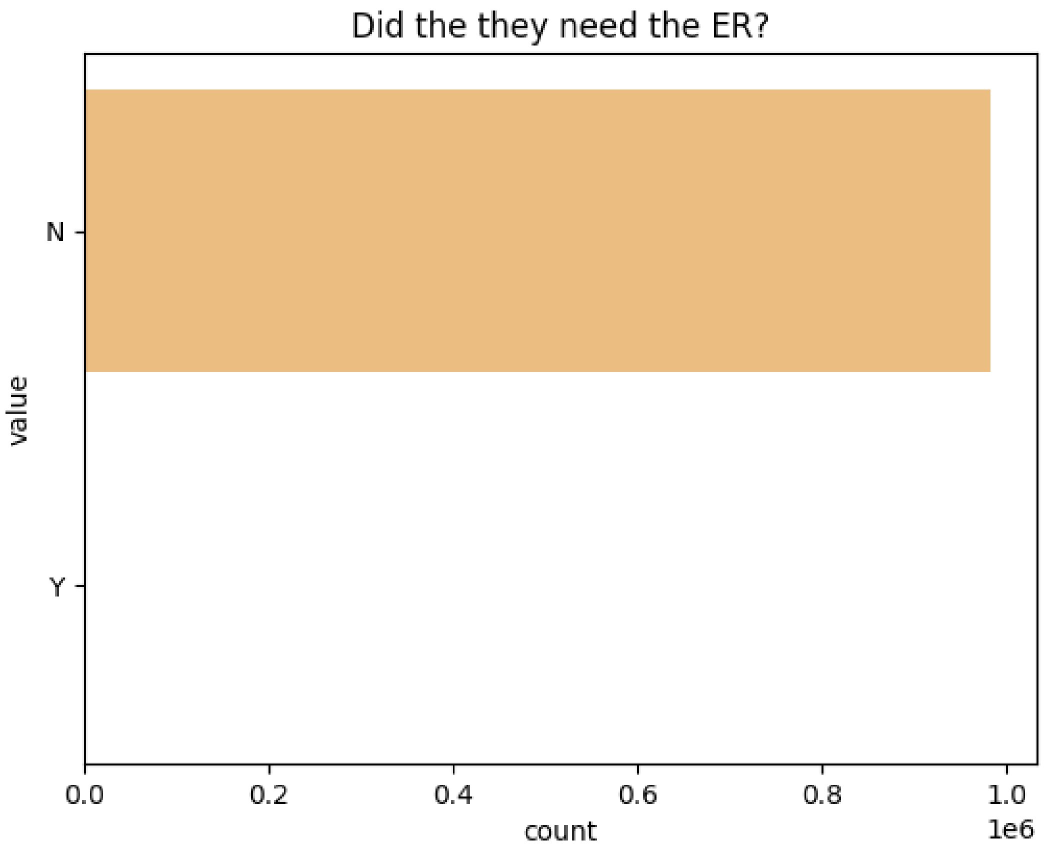


4.9.2 ER_VISIT column

[82]:
```
print('did the adverse reaction result in an ER Visit')
temp = break_down_columns(df,'ER_VISIT')
print('\n' + '-'*25 + '\n')

# temp.plot.
 ↪bar(title='ER_VISIT',x='value',y='count',rot=0,figsize=(5,5),width=0.
 ↪25,fontsize=18)
# temp = temp.set_index('value')
# temp.plot.pie(title='ER_VISIT',y='percent',figsize=(5, 5),autopct='%.
 ↪2f',fontsize=18)

# fig = px.bar(
#     temp,
#      x='count',
#      y='value',
#     title='Did the they need the ER?',
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="count",
    y="value",
    # hue="PERCENT"
    palette = "Spectral"
    ).set(title='Did the they need the ER?')
```

```
did the adverse reaction result in an ER Visit

break down of ER_VISIT

      column value   count  percent
0   ER_VISIT     N  983165    99.99
1   ER_VISIT     Y     140     0.01


-------------------------
```

```
[82]: [Text(0.5, 1.0, 'Did the they need the ER?')]
```



4.9.3 L_THREAT column

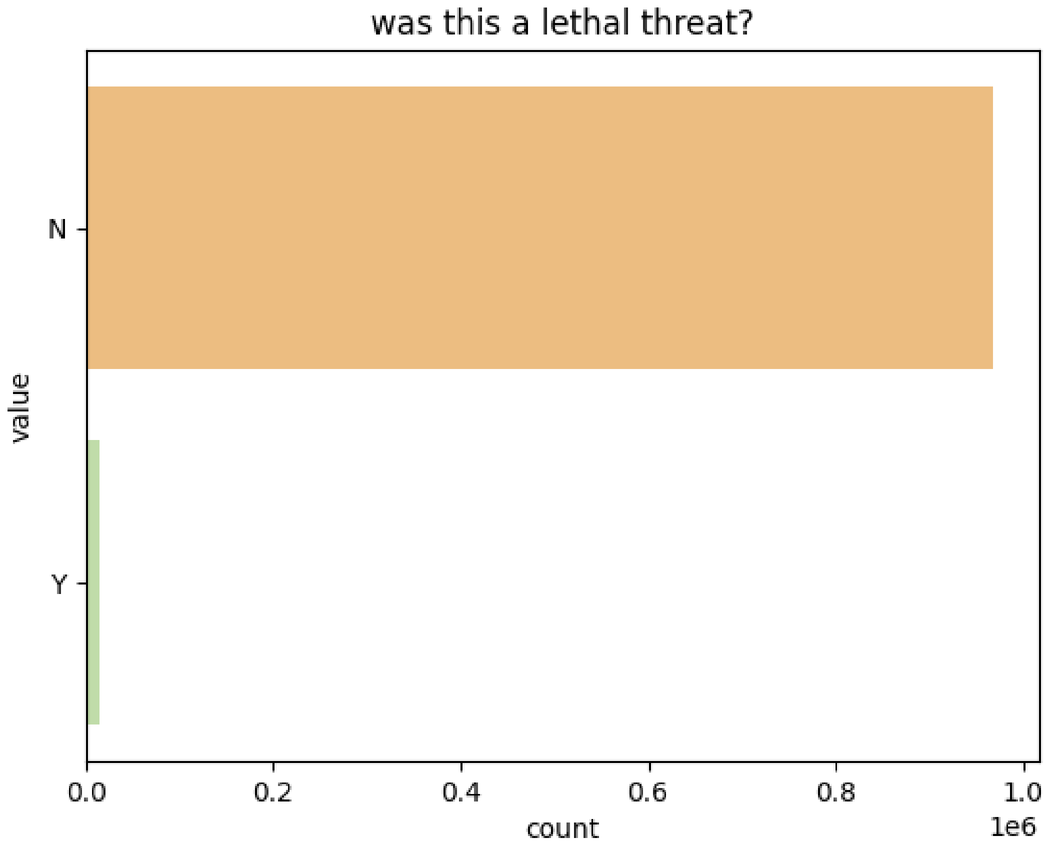
[83]:

```
print('Life Threatening/Leathal Threat')
temp = break_down_columns(df,'L_THREAT')
print('\n' + '-'*25 + '\n')

# temp.plot.
 ↪bar(title='L_THREAT',x='value',y='count',rot=0,figsize=(5,5),width=0.
 ↪25,fontsize=18)
# temp = temp.set_index('value')
# temp.plot.pie(title='L_THREAT',y='percent',figsize=(5, 5),autopct='%.
 ↪2f',fontsize=18)

# fig = px.bar(
#     temp,
#     x='count',
#     y='value',
#     title="was this a lethal threat?"
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="count",
    y="value",
    # hue="PERCENT"
    palette = "Spectral"
    ).set(title='was this a lethal threat?')
```

```
Life Threatening/Leathal Threat

break down of L_THREAT

     column value   count  percent
0  L_THREAT     N  967762    98.42
1  L_THREAT     Y   15543     1.58


-------------------------
```

[83]:

```
[Text(0.5, 1.0, 'was this a lethal threat?')]
```

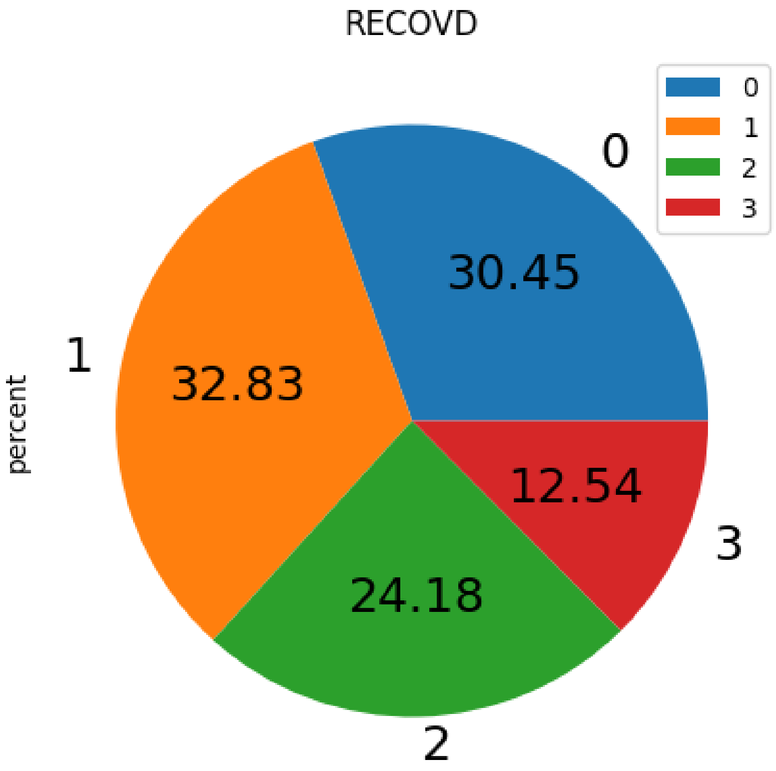


4.9.4 RECOVD column

[84]:
```
print('did the patient recover?')
temp = break_down_columns(df,'RECOVD')

print('\n' + '-'*25 + '\n')

# temp.plot.bar(title='RECOVD',x='value',y='count',rot=0,figsize=(5,5),width=0.
↪25,fontsize=18)
# temp = temp.set_index('value')
temp.plot.pie(title='RECOVD',y='percent',figsize=(5, 5),autopct='%.
↪2f',fontsize=18)

# fig = px.bar(
#     temp,
#     x='count',
#     y='value',
#     title="Did the patient recover? (Y=yes, N=no, U=unknown, nan=N/A )"
#     )
# fig.show()
```

```
did the patient recover?

break down of RECOVD

   column value   count  percent
0  RECOVD     Y  299382    30.45
1  RECOVD     N  322851    32.83
2  RECOVD     U  237799    24.18
3  RECOVD   nan  123273    12.54


-------------------------
```

```
[84]: <Axes: title={'center': 'RECOVD'}, ylabel='percent'>
```

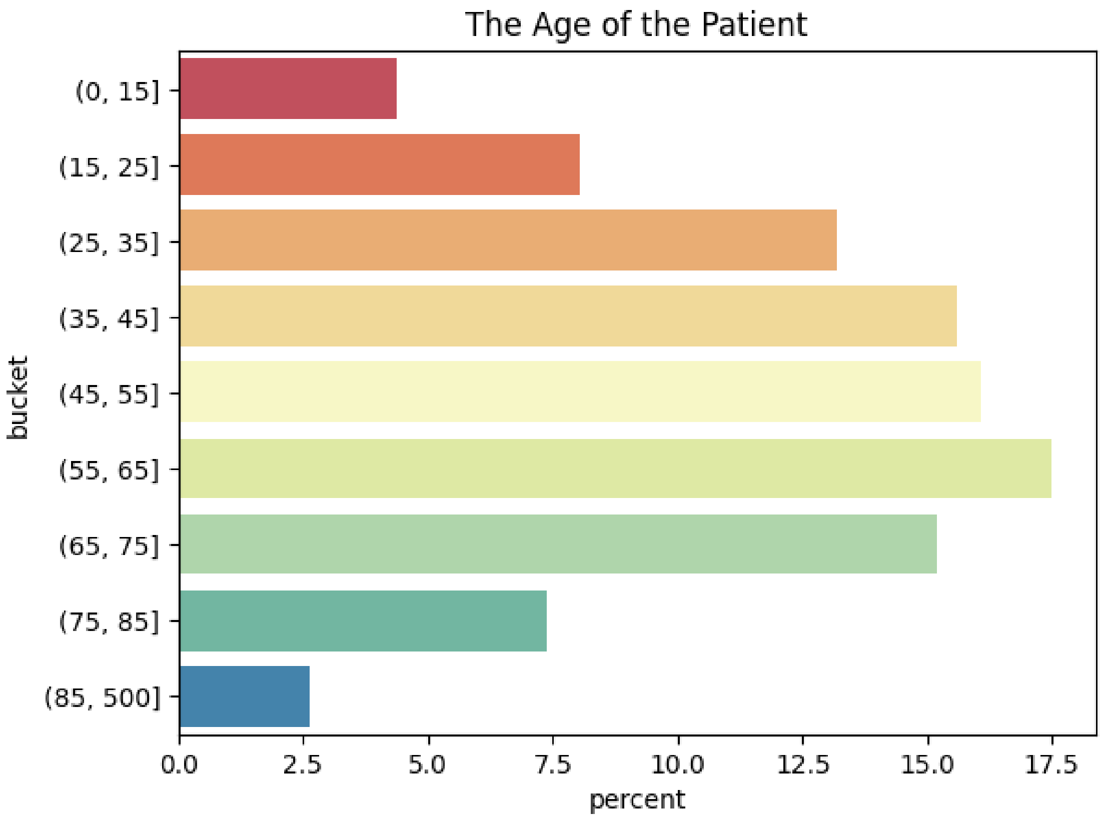


4.9.5 the Age of the patient

[85]:

```
print('the Age of the patient')
temp = break_down_buckets(df,'AGE_YRS',[0,15,25,35,45,55,65,75,85,500])
temp = temp.reset_index()
temp["bucket"] = temp["bucket"].astype(str)

# temp = temp.reset_index()
# temp.plot.
 ↪bar(title='AGE_YRS',x='bucket',y='AGE_YRS',rot=90,figsize=(5,5),width=0.
 ↪25,fontsize=18)
# # temp.plot.
 ↪bar(title='AGE_YRS',x='bucket',y='percent',rot=90,figsize=(5,5),width=0.
 ↪25,fontsize=18)

# fig = px.bar(
#     temp,
#     x='percent',
#     y='bucket',
#     title="The Age of the Patient"
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="percent",
    y="bucket",
    palette = "Spectral"
    ).set(title='The Age of the Patient')

print('\n' + '-'*25 + '\n')
```

```
the Age of the patient

column:  AGE_YRS
buckets:  [0, 15, 25, 35, 45, 55, 65, 75, 85, 500]

           AGE_YRS  percent
bucket
(0, 15]      38797     4.36
(15, 25]     71592     8.05
(25, 35]    117340    13.20
(35, 45]    138581    15.59
(45, 55]    142923    16.07
(55, 65]    155638    17.50
(65, 75]    135097    15.19
(75, 85]     65641     7.38
(85, 500]    23511     2.64
-------------------------
```

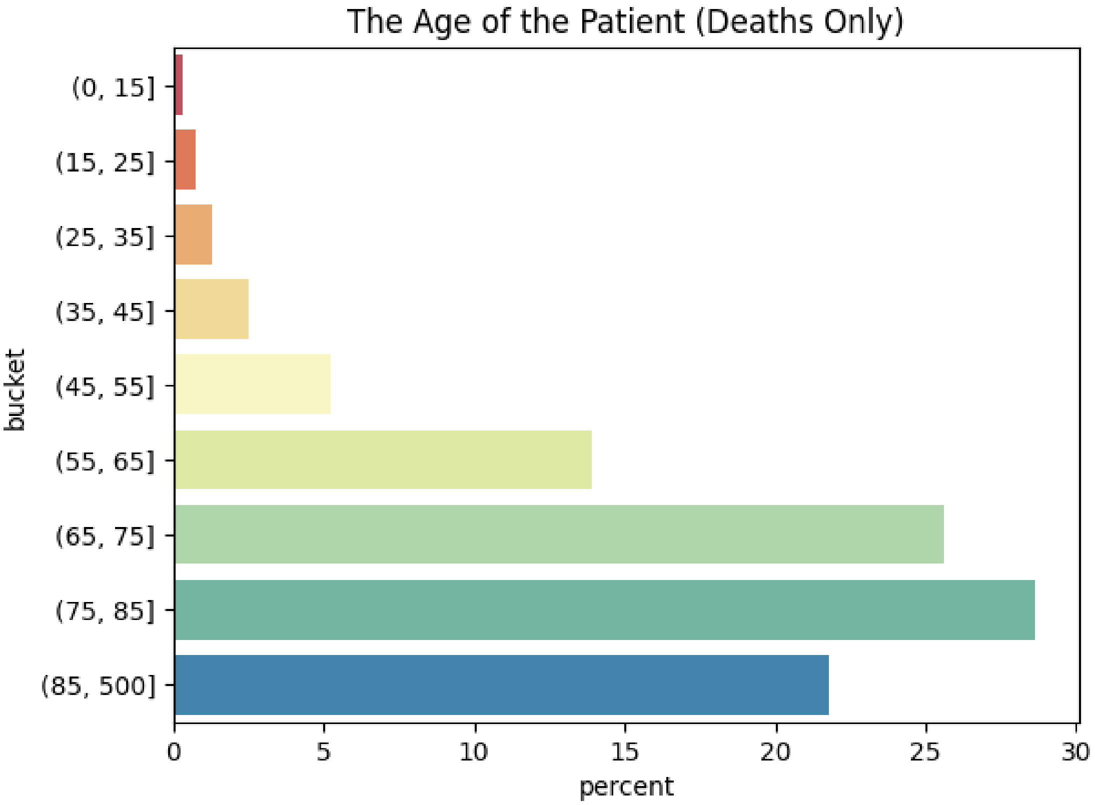


[86]:

```
temp =␣
 ↪break_down_buckets(df_death,'AGE_YRS',[0,15,25,35,45,55,65,75,85,500],message='***deaths␣
 ↪only***')
temp = temp.reset_index()
temp["bucket"] = temp["bucket"].astype(str)

# temp = temp.reset_index()
# temp.plot.bar(title='AGE_YRS (death␣
 ↪only)',x='bucket',y='AGE_YRS',rot=90,figsize=(5,5),width=0.25,fontsize=18)
# # temp.plot.bar(title='AGE_YRS (death␣
 ↪only)',x='bucket',y='percent',rot=90,figsize=(5,5),width=0.25,fontsize=18)

# fig = px.bar(
#     temp,
#     x='percent',
#     y='bucket',
#     title="The Age of the Patient"
```

```python
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="percent",
    y="bucket",
    palette = "Spectral"
    ).set(title='The Age of the Patient (Deaths Only)')

print('\n' + '-'*25 + '\n')
```

```
 ***deaths only***
column:  AGE_YRS
buckets:  [0, 15, 25, 35, 45, 55, 65, 75, 85, 500]

           AGE_YRS  percent
bucket
(0, 15]         54     0.29
(15, 25]       139     0.76
(25, 35]       233     1.27
(35, 45]       462     2.51
(45, 55]       963     5.23
(55, 65]      2558    13.90
(65, 75]      4710    25.60
(75, 85]      5272    28.66
(85, 500]     4007    21.78


-------------------------
```

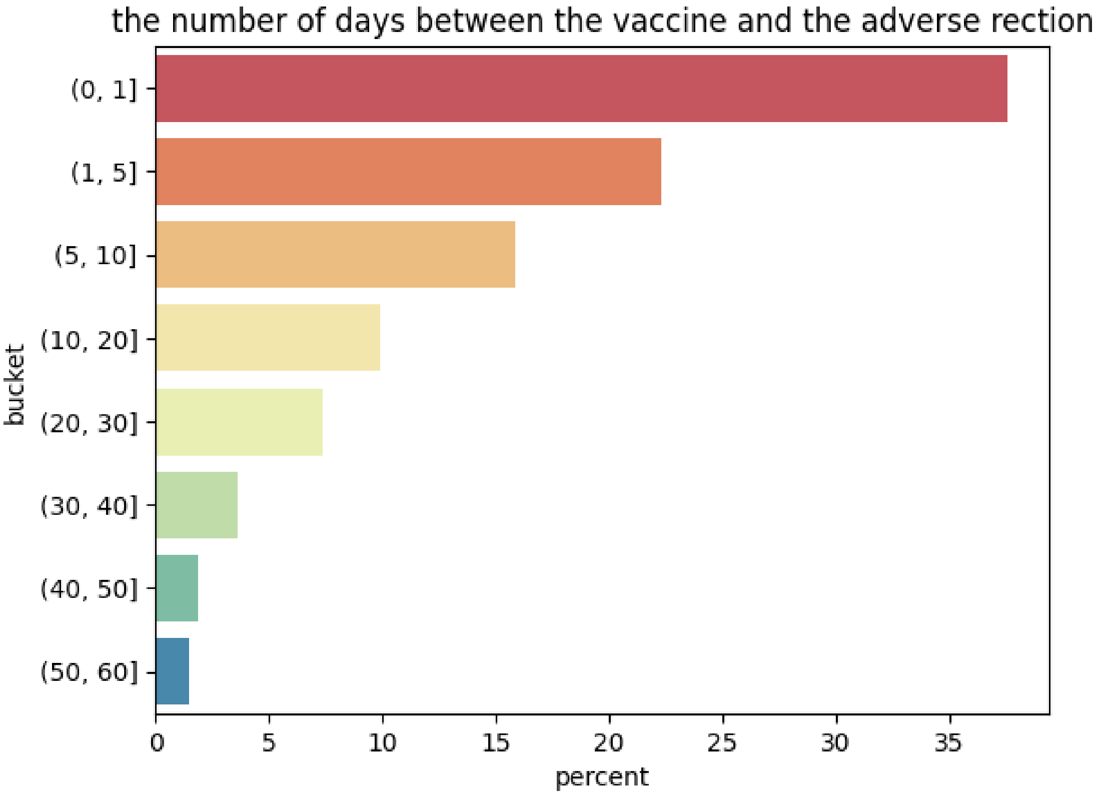


4.9.6 the number of days between the vaccine and the adverse rection

[87]:
```
print('the number of days between the vaccine and the adverse rection')
temp = break_down_buckets(df,'NUMDAYS',[0,1,5,10,20,30,40,50,60])
temp = temp.reset_index()
temp["bucket"] = temp["bucket"].astype(str)

# temp.plot.
 ↪barh(title='NUMDAYS',x='bucket',y='NUMDAYS',rot=0,figsize=(10,5),width=0.
 ↪25,fontsize=18)
# # temp.plot.
 ↪bar(title='AGE_YRS',x='bucket',y='percent',rot=90,figsize=(5,5),width=0.
 ↪25,fontsize=18)

# fig = px.bar(
#     temp,
#     x='percent',
#     y='bucket',
#     title="the number of days between the vaccine and the adverse rection"
#     )
# fig.show()
```

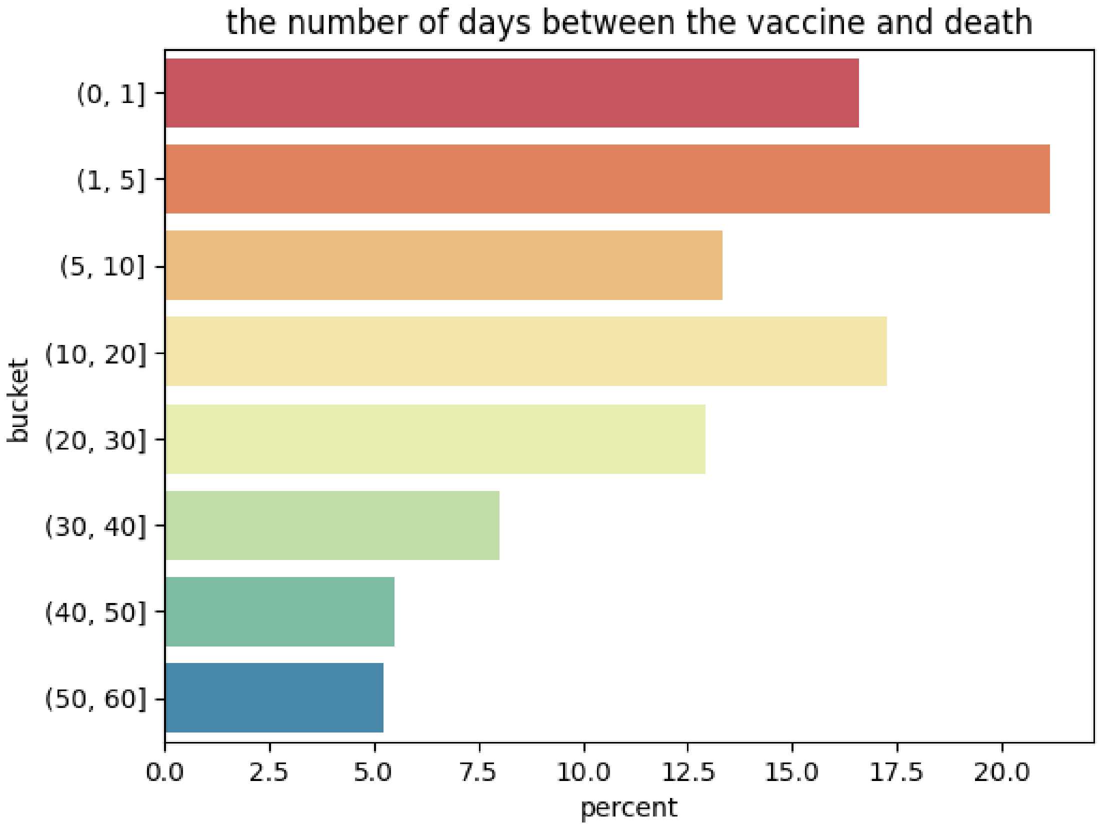
```
sns.barplot(
    data=temp,
    x="percent",
    y="bucket",
    palette = "Spectral"
    ).set(title='the number of days between the vaccine and the adverse␣
 ↪rection')

print('\n' + '-'*25 + '\n')
```

```
the number of days between the vaccine and the adverse rection

column:  NUMDAYS
buckets:  [0, 1, 5, 10, 20, 30, 40, 50, 60]

          NUMDAYS  percent
bucket
(0, 1]     147363    37.54
(1, 5]      87665    22.33
(5, 10]     62496    15.92
(10, 20]    38932     9.92
(20, 30]    28835     7.35
(30, 40]    14159     3.61
(40, 50]     7293     1.86
(50, 60]     5796     1.48


-------------------------
```



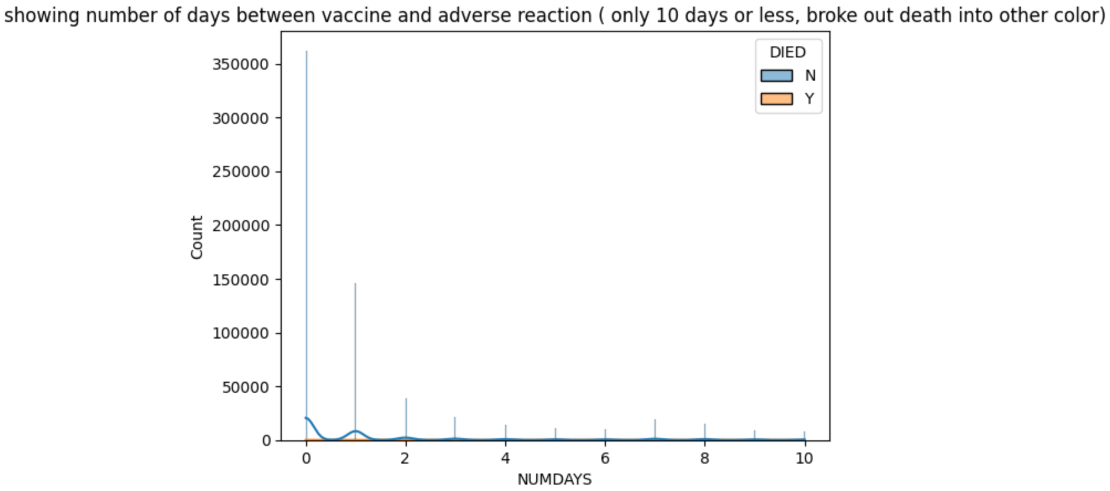
[88]:
```
temp =␣
  ↪break_down_buckets(df_death,'NUMDAYS',[0,1,5,10,20,30,40,50,60],message='***deaths␣
  ↪only***')
temp = temp.reset_index()
temp["bucket"] = temp["bucket"].astype(str)

# temp.plot.barh(title='NUMDAYS (deaths␣
  ↪only)',x='bucket',y='NUMDAYS',rot=0,figsize=(10,5),width=0.25,fontsize=18)
# temp.plot.
  ↪bar(title='AGE_YRS',x='bucket',y='percent',rot=90,figsize=(5,5),width=0.
  ↪25,fontsize=18)

# fig = px.bar(
#     temp,
#     x='percent',
#     y='bucket',
#     title="the number of days between the vaccine and death"
#     )
# fig.show()

sns.barplot(
    data=temp,
    x="percent",
    y="bucket",
    palette = "Spectral"
    ).set(title='the number of days between the vaccine and death')

print('\n' + '-'*25 + '\n')
```

```
 ***deaths only***
column:  NUMDAYS
buckets:  [0, 1, 5, 10, 20, 30, 40, 50, 60]

          NUMDAYS  percent
bucket
(0, 1]       1261    16.59
(1, 5]       1608    21.15
(5, 10]      1015    13.35
(10, 20]     1312    17.26
(20, 30]      984    12.94
(30, 40]      610     8.02
(40, 50]      417     5.48
(50, 60]      396     5.21


-------------------------
```



[89]:
```
temp = df[df['NUMDAYS'] <= 10]

sns.histplot(
    data=temp,
    x="NUMDAYS",
    hue='DIED',
    kde=True
    ).set(title='showing number of days between vaccine and adverse reaction (␣
↪only 10 days or less, broke out death into other color)')

# fig = px.histogram(
#     temp,
#     x="NUMDAYS",
#     color='DIED',
#     marginal="violin",
#     hover_data=temp.columns,
#     title="showing number of days between vaccine and adverse reaction ( only␣
↪25 days or less, 25% sample,broke out death into other color)"
#     )
# fig.update_layout(barmode='overlay')
# fig.update_traces(opacity=0.75)
# fig.show()
```

```
[89]: [Text(0.5, 1.0, 'showing number of days between vaccine and adverse reaction (
      only 10 days or less, broke out death into other color)')]
```



4.9.7 break down of the VAX_NAME column

```
[90]: print('break down of the VAX_NAME column')
      df_VN = break_down_columns(df,'VAX_NAME')

      print('\n***deaths only***')
      df_DVN = break_down_columns(df_death,'VAX_NAME')

      # df_CVN = df_VN.join(df_DVN[['value','count']],on='value')
      df_DVN = df_DVN.rename(columns={"count": "death_count"})


      print('calculating the death ratio ( death_count / count )')
      df_CVN = pd.concat([df_VN,df_DVN],keys=['value'],join="inner",axis=1)

      df_CVN = pd.merge(df_VN,df_DVN,on=['value'])
      df_CVN = df_CVN.drop(columns=['percent_x','percent_y','column_y'])
      df_CVN = df_CVN.rename(columns={"column_x": "column"})
      df_CVN['death_ratio'] = df_CVN['death_count'] / df_CVN['count']
      df_CVN['death_percent'] = df_CVN['death_ratio']*100

      display(df_CVN)
```

```python
print('\n' + '-'*25 + '\n')
```

```
break down of the VAX_NAME column

break down of VAX_NAME

     column                                         value   count  percent
0  VAX_NAME           COVID19 (COVID19 (PFIZER-BIONTECH))  447882    45.55
1  VAX_NAME                   COVID19 (COVID19 (UNKNOWN))    4999     0.51
2  VAX_NAME                   COVID19 (COVID19 (MODERNA))  443030    45.06
3  VAX_NAME          COVID19 (COVID19 (MODERNA BIVALENT))    6745     0.69
4  VAX_NAME  COVID19 (COVID19 (PFIZER-BIONTECH BIVALENT))    8890     0.90
5  VAX_NAME                   COVID19 (COVID19 (JANSSEN))   71550     7.28
6  VAX_NAME                   COVID19 (COVID19 (NOVAVAX))     209     0.02


***deaths only***

break down of VAX_NAME

     column                                         value  count  percent
0  VAX_NAME                   COVID19 (COVID19 (MODERNA))   8666    43.71
1  VAX_NAME           COVID19 (COVID19 (PFIZER-BIONTECH))   9196    46.38
2  VAX_NAME                   COVID19 (COVID19 (UNKNOWN))     66     0.33
3  VAX_NAME                   COVID19 (COVID19 (JANSSEN))   1781     8.98
4  VAX_NAME  COVID19 (COVID19 (PFIZER-BIONTECH BIVALENT))     75     0.38
5  VAX_NAME          COVID19 (COVID19 (MODERNA BIVALENT))     44     0.22
calculating the death ratio ( death_count / count )

     column                                         value   count  \
0  VAX_NAME           COVID19 (COVID19 (PFIZER-BIONTECH))  447882
1  VAX_NAME                   COVID19 (COVID19 (UNKNOWN))    4999
2  VAX_NAME                   COVID19 (COVID19 (MODERNA))  443030
3  VAX_NAME          COVID19 (COVID19 (MODERNA BIVALENT))    6745
4  VAX_NAME  COVID19 (COVID19 (PFIZER-BIONTECH BIVALENT))    8890
5  VAX_NAME                   COVID19 (COVID19 (JANSSEN))   71550

   death_count  death_ratio  death_percent
0         9196         0.02           2.05
1           66         0.01           1.32
2         8666         0.02           1.96
3           44         0.01           0.65
4           75         0.01           0.84
5         1781         0.02           2.49


-------------------------
```

4.10 Approximating actual adverse reaction numbers

VAERS only contains reported data and
'…fewer than 1% of vaccine adverse events are reported.'
Source: https://digital.ahrq.gov/sites/default/files/docs/publication/r18hs017045-lazarus-final-report-2011.pdf (page 6)

we will me multiplying the counts by 80 and 120,
in order to get an approzimate min and max of what the numbers might actually be.

[91]:

```
def print_percent(vmin,vmax,label0,vcount,label1):
    """
    vmin: vaers min
    vmax: vaers max
    label0: vaers label
    vcount: vaxxed count
    label1: vaxxed label
    """
    print(
        '( {label0} / {lavel1} ) * 100\n'.format(
            label0=label0,
            lavel1=label1
            ),
        'min: ( {0:,} / {1:,} ) * 100 \n'.format(vmin,vcount),
        'max: ( {0:,} / {1:,} ) * 100 \n'.format(vmax,vcount),
        '{:.2f} %'.format((vmin/vcount)*100),
        '-',
        '{:.2f} %'.format((vmax/vcount)*100),
        '\n'
        )

print_percent(
    len(df)*80,
    len(df)*120,
    'approx adverse reactions',
    vaxx,
    'vaxxed [1 or more shots]'
    )

print_percent(
    len(df_death)*80,
    len(df_death)*120,
    'approx adverse deaths',
    vaxx,
    'vaxxed [1 or more shots]'
    )

df_nrecovd = df[df['RECOVD']=='N']
```

```
print_percent(
    len(df_nrecovd)*80,
    len(df_nrecovd)*120,
    'approx no recovery',
    vaxx,
    'vaxxed [1 or more shots]'
    )

df_urecovd = df[df['RECOVD']=='U']
print_percent(
    (len(df_nrecovd) + (len(df_urecovd)*0.5))*80,
    (len(df_nrecovd) + (len(df_urecovd)*0.5))*120,
    'approx no recovery + (50% of unknowns)',
    vaxx,
    'vaxxed [1 or more shots]'
    )
```

```
( approx adverse reactions / vaxxed [1 or more shots] ) * 100
 min: ( 78,664,400 / 500,864,529 ) * 100
 max: ( 117,996,600 / 500,864,529 ) * 100
 15.71 % - 23.56 %

( approx adverse deaths / vaxxed [1 or more shots] ) * 100
 min: ( 1,586,240 / 500,864,529 ) * 100
 max: ( 2,379,360 / 500,864,529 ) * 100
 0.32 % - 0.48 %

( approx no recovery / vaxxed [1 or more shots] ) * 100
 min: ( 25,828,080 / 500,864,529 ) * 100
 max: ( 38,742,120 / 500,864,529 ) * 100
 5.16 % - 7.74 %

( approx no recovery + (50% of unknowns) / vaxxed [1 or more shots] ) * 100
 min: ( 35,340,040.0 / 500,864,529 ) * 100
 max: ( 53,010,060.0 / 500,864,529 ) * 100
 7.06 % - 10.58 %
```

4.11 Women's Reproductive Symptoms

Why? I have women in my life that were curious about this.

WRS_list (list) : a list of symptoms that effect or could cause effects to a women's reproductive system
df_WRS (DataFrame) : a dataframe that contains VAERS events that have at least 1 of the WRS symptoms

[92]:

```
def symptom_filter_search(idf, search_list):
    """
    returns a dataframe pf vaers events where the patient
    has had 1 or more of the symptoms on the list
    """
    print('this could take between 20sec-60sec')

    data = idf.to_dict(orient='records')

    search_list = [i.upper() for i in search_list]

    results = []
    for index,d in enumerate(data):
        # if index%5000 == 0:
        #     print('{:.2f}'.format(index/len(data)))
        try:
            d['SYMPTOMS'] = [i.upper() for i in d['SYMPTOMS'] if isinstance(i,␣
 ↪str)]

            symptom_match = list(set(d['SYMPTOMS']) & set(search_list))
            d['SYMPTOMS_MATCH'] = symptom_match
            d['SYMPTOMS_MATCH_LENGTH'] = len(symptom_match)
            if len(symptom_match) > 0:
                results.append(d)
        except:
            pass

    return pd.DataFrame(results)
```

[93]:

```
WRS_list = [
    'Intermenstrual bleeding',
    'Menopause',
    'Heavy menstrual bleeding',
    'dysmenorrhoea',
    'ABNORMAL UTERINE BLEEDING',
    'MATERNAL EXPOSURE BEFORE PREGNANCY',
    'MENSTRUATION IRREGULAR',
    'Oligomenorrhea',
    'OLIGOMENORRHOEA',
    'POLYMENORRHOEA',
    'MENSTRUAL DISORDER',
    'OLIGOMENORRHOEA',
    'ANOVULATORY CYCLE',
    'OVULATION DELAYED',
    'BACTERIAL VAGINOSIS',
    'GYNAECOLOGICAL EXAMINATION ABNORMAL',
    'OVARIAN CYST',
    'BIOPSY UTERUS',
    'UTERINE LEIOMYOMA',
    'HOT FLUSH',
    'BREAST TENDERNESS',
    'BREAST SWELLING',
    'BREAST PAIN',
    'VAGINAL HAEMORRHAGE'
    ]
WRS_list = [i.upper() for i in WRS_list]
```

[94]:

```
df_WRS = symptom_filter_search(df,WRS_list)

print('df_WRS.head(5)')
display(df_WRS.head(5))
```

```
this could take between 20sec-60sec
df_WRS.head(5)

   VAERS_ID    RECVDATE STATE  AGE_YRS  CAGE_YR  CAGE_MO SEX RPT_DATE  \
0    902796  12/16/2020    TX    34.00    34.00      NaN   F      NaN
1    903202  12/17/2020    NE    41.00    41.00      NaN   F      NaN
2    903247  12/17/2020    AR    42.00    42.00      NaN   F      NaN
3    903329  12/18/2020    PA    46.00    46.00      NaN   F      NaN
4    903345  12/18/2020    VA    37.00    37.00      NaN   F      NaN

                                                                                              ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                             ␣
 ↪                                                                                     ␣
 ↪SYMPTOM_TEXT  \
0  I inserted my NuvaRing birth control on 12/14/2020.  I have the Covid19␣
 ↪vaccine on 12/15/2020 at about 8:30am.  The injection site was mildly sore and␣
 ↪that continued into the next day.  I woke up on 12/16/2020 feeling a little␣
 ↪off and it progressed throughout the day.  I felt a headache that I knew was␣
 ↪turning into a migraine, threw up a few times (this is normal for me when I␣
 ↪get migraines), and started having hot flashes.  My temperature never went␣
 ↪above 98.  After sleeping a few hours and taking some Excedrin migraine I was␣
 ↪feeling much better, though still a little sickly, by 4pm.  I really think␣
 ↪this is due to my migraines (which I get roughly once a month) even though it␣
 ↪was a day later than normal.
```

```
1                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                          Severe dizzy spell about 5-10␣
 ↪following injection, helped to the floor, this lasted approximately 30-40␣
 ↪minutes after start of symptom.  Hot flashes and visual disturbance lasting␣
 ↪into following day.
2                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪         I had arm pain. Yesterday I had a hot flash with throwing up and␣
 ↪diarrhea.
3                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                   hot flushing feeling, light␣
 ↪headed, legs heavy gave patient a chair to sit and candy symptoms resolved by␣
 ↪11:36
4                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                                                               ␣
 ↪                                         Right arm soreness, headache, hot␣
 ↪flashes.

  DIED DATEDIED L_THREAT ER_VISIT HOSPITAL  HOSPDAYS X_STAY DISABLE RECOVD  \
0    N      NaN        N        N        N       NaN    NaN     NaN      Y
1    N      NaN        N        N        N       NaN    NaN     NaN      Y
2    N      NaN        N        N        N       NaN    NaN     NaN      N
3    N      NaN        N        N        N       NaN    NaN     NaN      Y
4    N      NaN        N        N        N       NaN    NaN     NaN      Y
```

```
     VAX_DATE  ONSET_DATE  NUMDAYS LAB_DATA V_ADMINBY V_FUNDBY  \
0  12/15/2020  12/16/2020     1.00      NaN       PVT      NaN
1  12/16/2020  12/16/2020     0.00     None       OTH      NaN
2  12/15/2020  12/16/2020     1.00       No       UNK      NaN
3  12/17/2020  12/17/2020     0.00      NaN       PVT      NaN
4  12/16/2020  12/17/2020     1.00      NaN       WRK      NaN

                                                                                ␣
 ↪                                                                              ␣
 ↪                                                                        ␣
 ↪OTHER_MEDS  \
0                                                                               ␣
 ↪                                                                              ␣
 ↪                                      Methylphenidate Citalopram Buspirone␣
 ↪Tylenol
1  Buspar, propranolol,  escitalopram, crestor, amitriptyline,  baby aspirin,␣
 ↪super B complex, black elderberry, calcium/mag/zinc supplement, CoQ10, vitamin␣
 ↪D3, biotin, antarctic Krill oil, women's multivitamin,  tart cherry␣
 ↪supplement, ibupr
2                                                                               ␣
 ↪                                                                         I␣
 ↪take Estradiol, Metprolol, Lexapro, Multivitamin, Calcium, Ambien, Benadryl,␣
 ↪Melatonin
3                                                                               ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪NaN
4                                                                               ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪NaN

  CUR_ILL  \
0    None
1    None
2      No
3     NaN
4    None

                                                                                ␣
 ↪                                                                              ␣
 ↪                HISTORY  \
0  I normally get a migraine a day or two after I insert my NuvaRing.  I␣
 ↪inserted it on Monday, 12/14/2020 and I think all my symptoms are related to␣
 ↪that rather than the vaccination.
```

```
1                                                                               ␣
 ↪                                                        Migraine,  anxiety,␣
 ↪depression, hyperlipidemia
2                                                                               ␣
 ↪                                                                               ␣
 ↪             I have SVT
3                                                                               ␣
 ↪                                                                               ␣
 ↪                    NaN
4                                                                               ␣
 ↪                                                                               ␣
 ↪                   None

  PRIOR_VAX SPLTTYPE  FORM_VERS TODAYS_DATE BIRTH_DEFECT OFC_VISIT  \
0       NaN      NaN          2  12/16/2020          NaN       NaN
1       NaN      NaN          2  12/17/2020          NaN       NaN
2       NaN    vsafe          2  12/17/2020          NaN       NaN
3       NaN      NaN          2  12/18/2020          NaN       NaN
4       NaN      NaN          2  12/18/2020          NaN       NaN

  ER_ED_VISIT                 ALLERGIES VAX_TYPE         VAX_MANU VAX_LOT  \
0         NaN                      None  COVID19  PFIZER\BIONTECH  EH9899
1         NaN                   Aimovig  COVID19  PFIZER\BIONTECH  EH9899
2         NaN  I am allergic to Sulphur  COVID19  PFIZER\BIONTECH     NaN
3         NaN                       NaN  COVID19  PFIZER\BIONTECH  EK5730
4         NaN                      None  COVID19  PFIZER\BIONTECH     NaN

  VAX_DOSE_SERIES VAX_ROUTE VAX_SITE                             VAX_NAME  \
0               1       SYR       LA  COVID19 (COVID19 (PFIZER-BIONTECH))
1               1        IM       RA  COVID19 (COVID19 (PFIZER-BIONTECH))
2             UNK        IM       LA  COVID19 (COVID19 (PFIZER-BIONTECH))
3               1        IM       RA  COVID19 (COVID19 (PFIZER-BIONTECH))
4             UNK       NaN       RA  COVID19 (COVID19 (PFIZER-BIONTECH))

                                                                                ␣
 ↪                SYMPTOMS  \
0  [CONDITION AGGRAVATED, FEELING ABNORMAL, HOT FLUSH, INJECTION SITE PAIN,␣
 ↪MALAISE, MIGRAINE, VOMITING]
1                                                              [DIZZINESS, HOT␣
 ↪FLUSH, VISUAL IMPAIRMENT]
2                                                    [DIARRHOEA, HOT FLUSH, PAIN␣
 ↪IN EXTREMITY, VOMITING]
3                                                                [DIZZINESS, HOT␣
 ↪FLUSH, LIMB DISCOMFORT]
4                                                               [HEADACHE, HOT␣
 ↪FLUSH, PAIN IN EXTREMITY]
```

```
   COVID_VAX SYMPTOMS_MATCH  SYMPTOMS_MATCH_LENGTH
0          1    [HOT FLUSH]                      1
1          1    [HOT FLUSH]                      1
2          1    [HOT FLUSH]                      1
3          1    [HOT FLUSH]                      1
4          1    [HOT FLUSH]                      1
```

4.11.1 What percent of women experienced WRS during their adverse reactions?

(women experiencing reproductive symptoms   number of women in VAERS data) * 100

WRS_ratio (float) : count WRS / count VAERS

[95]:

```
#get count of women in VAERS data

w_df = df[df['SEX']=='F']
u_df = df[df['SEX']=='U']
w_count = len(w_df) + (len(u_df)/2) #half of unknown

WRS_ratio = (len(df_WRS)/w_count)

print( '{:.2f} %'.format(WRS_ratio*100))

file_name = os.path.join(DATAPATH,'WRS.csv')
df_WRS.to_csv(file_name)

print('VAERS records of women experiencing reproductive symptoms have been␣
 ↪saved.')
print('saved: ',file_name)
```

```
3.35 %
VAERS records of women experiencing reproductive symptoms have been saved.
saved:  .\AllVAERSDataCSVS\WRS.csv
```

4.11.2 Approximate the number of WRS in reality

since '~63% of the people who are vaccinated are women' source: https://www.statista.com/statistics/1212103/share-of-persons-initiating-covid-vaccinations-by-gender-us-first-month/

we will be multiplying the total number of people vaccinated (1 or more shots) by 0.63 to get the count of women vaccinated.

women_vaxx (float) : an approximate number of women who have had 1 or more vaccine shot.

> note: 0.63 is an estimate, and the actual could be somewhere between 0.50 and 0.70

[96]:

```
women_vaxx = vaxx * 0.63
print(women_vaxx)
```

```
315544653.27
```

4.11.3 approximate real WRS

if we assume that the VAERS data is a random sample (or close to it)
then the ratio of WRS systems should be the same…

and thus we can get an approximate number of women that would be experiencing reproductive symptoms by multiplying the number of vaxxed women by the ratio

WRS (float) : the approximate number of actual women experiencing reproductive symptoms min_WRS (float) : WRS * 0.80 max_WRS (float) : WRS * 1.20

[97]:

```
WRS = women_vaxx * WRS_ratio
min_WRS = WRS*0.80
max_WRS = WRS*1.20
```

[98]:

```
cl = [35,5,15]
print_row(['total vaxxed (1 or more)','---','{:,.2f}'.
 ↪format(vaxx)],column_lengths=cl)
print_row(['women vaxxed ~0.63%','---','{:,.2f}'.
 ↪format(women_vaxx)],column_lengths=cl)
print_row(['repro sympt / women count','---','{:,.4f}'.
 ↪format(WRS_ratio)],column_lengths=cl)
print_row(['(repro sympt / women count) * 100','---','{:,.2f} %'.
 ↪format(WRS_ratio*100)],column_lengths=cl)
print_row(['women w/ repro symptoms','---','{:,.2f}'.
 ↪format(WRS)],column_lengths=cl)
print_row(['min women w/ repro symptoms','---','{:,.2f}'.
 ↪format(min_WRS)],column_lengths=cl)
print_row(['max women w/ repro symptoms','---','{:,.2f}'.
 ↪format(max_WRS)],column_lengths=cl)
```

```
total vaxxed (1 or more)            ---  500,864,529.00
women vaxxed ~0.63%                 ---  315,544,653.27
repro sympt / women count           ---  0.0335
(repro sympt / women count) * 100   ---  3.35 %
women w/ repro symptoms             ---  10,575,525.70
min women w/ repro symptoms         ---  8,460,420.56
max women w/ repro symptoms         ---  12,690,630.84
```

4.11.4 Where do the WRS (Women reproductive symptoms) rank against with the other symptoms?

[99]:

```
temp = []

for index,i in enumerate(all_symptoms):
    if i[0].upper() in WRS_list:
        temp.append(
            {
                'index':index,
                'symptoms':i[0],
                'count': '{:,.2f}'.format(i[1]),
                'percent': '{:.2f}'.format((i[1]/len(all_symptoms))*100)
            }
        )

temp = pd.DataFrame(temp)
display(temp)
```

```
    index                            symptoms     count percent
0     133             HEAVY MENSTRUAL BLEEDING  5,702.00   46.80
1     172              MENSTRUATION IRREGULAR  4,346.00   35.67
2     173                           HOT FLUSH  4,333.00   35.56
3     198                  MENSTRUAL DISORDER  3,285.00   26.96
4     248                       DYSMENORRHOEA  2,399.00   19.69
5     251                         BREAST PAIN  2,335.00   19.16
6     279                  VAGINAL HAEMORRHAGE  1,996.00   16.38
7     310             INTERMENSTRUAL BLEEDING  1,688.00   13.85
8     480                     BREAST SWELLING    910.00    7.47
9     497                      POLYMENORRHOEA    855.00    7.02
10    539                      OLIGOMENORRHOEA    769.00    6.31
11    598                   BREAST TENDERNESS    648.00    5.32
12    782  MATERNAL EXPOSURE BEFORE PREGNANCY    428.00    3.51
13   1272                         OVARIAN CYST    194.00    1.59
14   1286                           MENOPAUSE    192.00    1.58
15   1461                    UTERINE LEIOMYOMA    152.00    1.25
16   1674           ABNORMAL UTERINE BLEEDING    118.00    0.97
17   2987                 BACTERIAL VAGINOSIS     36.00    0.30
18   3149                    ANOVULATORY CYCLE     32.00    0.26
19   3374                       BIOPSY UTERUS     27.00    0.22
20   3667                   OVULATION DELAYED     22.00    0.18
21   5243 GYNAECOLOGICAL EXAMINATION ABNORMAL      9.00    0.07
```

[ ]:

[ ]: