Final_vdb_report

September 12, 2021

1 Vdb

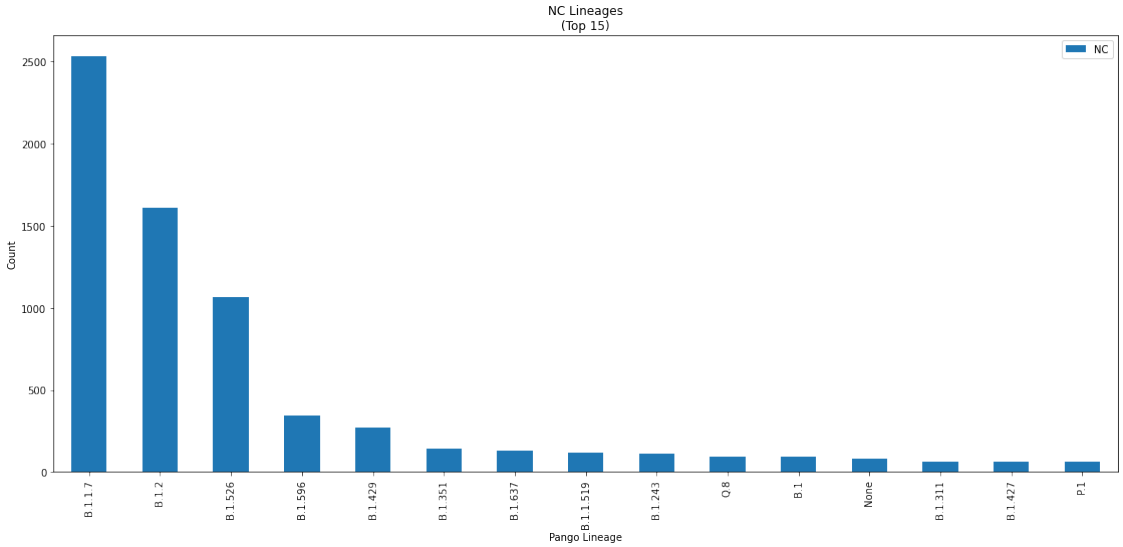
1.1 Lineages command

Searches the specified cluster (or all viruses if no cluster is given) for viruses belonging to the specified Pango lineage. **NC lineages from 09/01/20 - 05/15/21**

[1]:
```
#Used if only ONE State is being analyzed.
def graph_total_lineages(a_file):
    import pandas as pd

    graph_name= input("Where is this data from: (leave blank for default graph␣
 ↪naming)")

    #Read file and get line 1 of csv
    with open (a_file) as f:
        data= f.readlines()
        first_line=data[0].strip()
        f.close()


    '''
    #Pandas datafram of CSV.
    data=pd.read_csv(a_file, sep=",", header=0)

    #Instert headers

    new_data=data.rename(columns={first_line: graph_name})

    #Top 15 variants from file.
    top=new_data.head(15)
    '''

    if graph_name =="":
        #Pandas datafram of CSV.
        data=pd.read_csv(a_file, sep=",", header=0)

        #Instert headers
```

```
        new_data=data.rename(columns={first_line: "Query Count"})

        #Top 15 variants from file.
        top=new_data.head(15)

        top.plot(title='Total Count 09/20 - 05/21\n(Top 15)',ylabel="Count" ,␣
↪xlabel="Pango Lineage",kind="bar",figsize=(19, 8))
    else:
        #Pandas datafram of CSV.
        data=pd.read_csv(a_file, sep=",", header=0)

        #Instert headers

        new_data=data.rename(columns={first_line: graph_name})

        #Top 15 variants from file.
        top=new_data.head(15)
        print("\nDataFrame Used\n\n",top)
        top.plot(title= graph_name+' Lineages\n(Top 15)', ylabel="Count" ,␣
↪xlabel="Pango Lineage",kind="bar",figsize=(19, 8))

#Run this line to test Function above.
graph_total_lineages("01_lineages_nc_sep_to_may.csv")
```

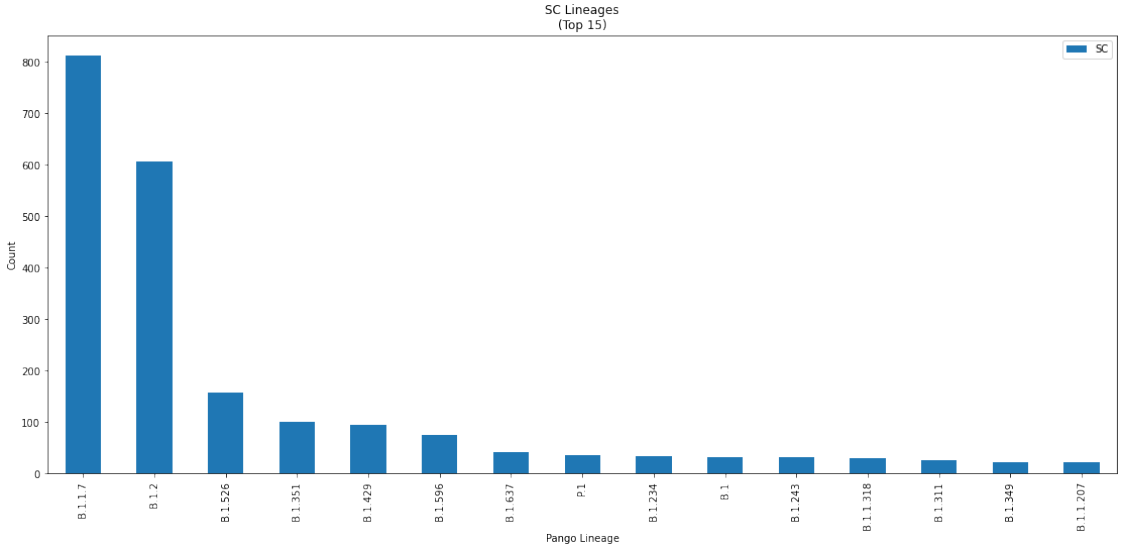
```
Where is this data from: (leave blank for default graph naming)NC

DataFrame Used

              NC
B.1.1.7     2532
B.1.2       1610
B.1.526     1065
B.1.596      342
B.1.429      270
B.1.351      141
B.1.637      131
B.1.1.519    116
B.1.243      111
Q.8           96
B.1           91
None          82
B.1.311       60
B.1.427       60
P.1           60
```



SC lineages from 09/01/20 - 05/15/21

[2]:
```
graph_total_lineages("01_lineages_sc_sep_to_may.csv")
```

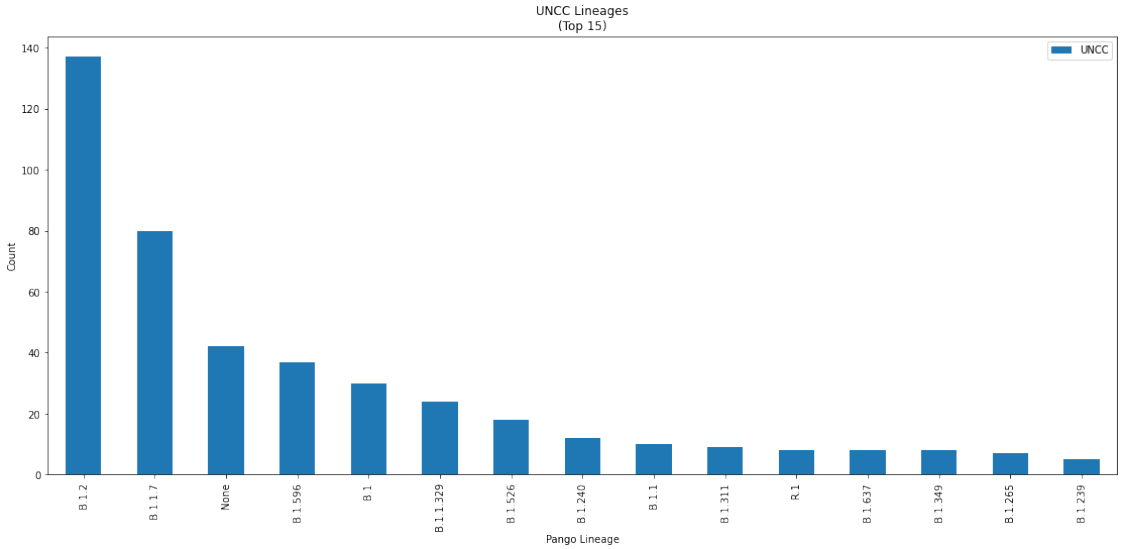
```
Where is this data from: (leave blank for default graph naming)SC

DataFrame Used

             SC
B.1.1.7     812
B.1.2       607
B.1.526     158
B.1.351     101
B.1.429      94
B.1.596      75
B.1.637      42
P.1          36
B.1.234      35
B.1          33
B.1.243      33
B.1.1.318    31
B.1.311      26
B.1.349      23
B.1.1.207    22
```



UNCC lineages from 09/01/20 - 05/15/21

[3]:
```
graph_total_lineages("01_lineages_UNCC.csv")
```

```
Where is this data from: (leave blank for default graph naming)UNCC

DataFrame Used

            UNCC
B.1.2        137
B.1.1.7       80
None          42
B.1.596       37
B.1           30
B.1.1.329     24
B.1.526       18
B.1.240       12
B.1.1         10
B.1.311        9
R.1            8
B.1.637        8
B.1.349        8
B.1.265        7
B.1.239        5
```

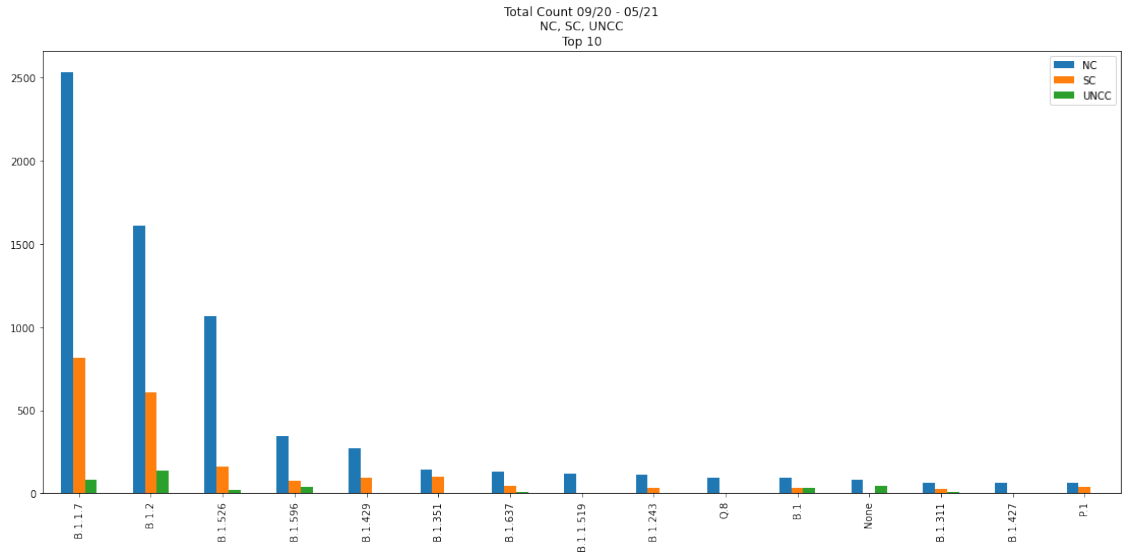


Combined lineages from 09/01/20 - 05/15/21

[4]:
```
#Function to get name of vdb generated first line.
def get_name(a_file):
    with open (a_file) as f:
        data = f.readlines()
        name=data[0].strip()
        f.close()
    return name

#Limited to three files because in this part of research
#Only data from NC, SC, and UNCC was used.
def multiple_lineage_graph(file_1, file_2, file_3):
    import pandas as pd
    #Get names for graphs
    file_names=[]
    for i in range(0,3):
        name=input("Enter Location of file number " +str(i+1)+ " Location (i.e␣
 ↪State Mame): ")
        file_names.append(name)

    #open First file, and look for line 1 tittle (given by vdb)
    #This will then be replaced with a different column name.
    lineage_1=pd.read_csv(file_1, sep=",", header=0)
    data=lineage_1.rename(columns={get_name(file_1): file_names[0]})
    top_file_1=data.head(15)

    #Repeat for file 2
    lineage_2=pd.read_csv(file_2, sep=",", header=0)
```

```
    data=lineage_2.rename(columns={get_name(file_2): file_names[1]})
    top_file_2=data.head(15)

    #Repeat for file 3
    lineage_3=pd.read_csv(file_3, sep=",", header=0)
    data=lineage_3.rename(columns={get_name(file_3): file_names[2]})
    top_file_3=data.head(15)

    #Join dataframes
    combined=top_file_1.join(top_file_2)
    final_combined= combined.join(top_file_3)
    print("\n\t      Dataframe Used\n\n",final_combined)

    final_combined.plot(title='Total Count 09/20 - 05/21\n'+  file_names[0]+',␣
 ↪'+file_names[1] +', '+
                        file_names[2]+'\nTop 10',kind="bar", figsize=(19, 8))




#Run this line to test
multiple_lineage_graph("01_lineages_nc_sep_to_may.
 ↪csv","01_lineages_sc_sep_to_may.csv","01_lineages_UNCC.csv")
```

```
Enter Location of file number 1 Location (i.e State Mame): NC
Enter Location of file number 2 Location (i.e State Mame): SC
Enter Location of file number 3 Location (i.e State Mame): UNCC

              Dataframe Used

              NC     SC   UNCC
B.1.1.7    2532  812.0   80.0
B.1.2      1610  607.0  137.0
B.1.526    1065  158.0   18.0
B.1.596     342   75.0   37.0
B.1.429     270   94.0    NaN
B.1.351     141  101.0    NaN
B.1.637     131   42.0    8.0
B.1.1.519   116    NaN    NaN
B.1.243     111   33.0    NaN
Q.8          96    NaN    NaN
B.1          91   33.0   30.0
None         82    NaN   42.0
B.1.311      60   26.0    9.0
B.1.427      60    NaN    NaN
P.1          60   36.0    NaN
```



1.2 Trends command

For the Pango lineages with the highest counts in specified cluster, this calculates how the fractions of these lineages have changed over time. **NC Trends from 09/01/20 - 05/15/21**

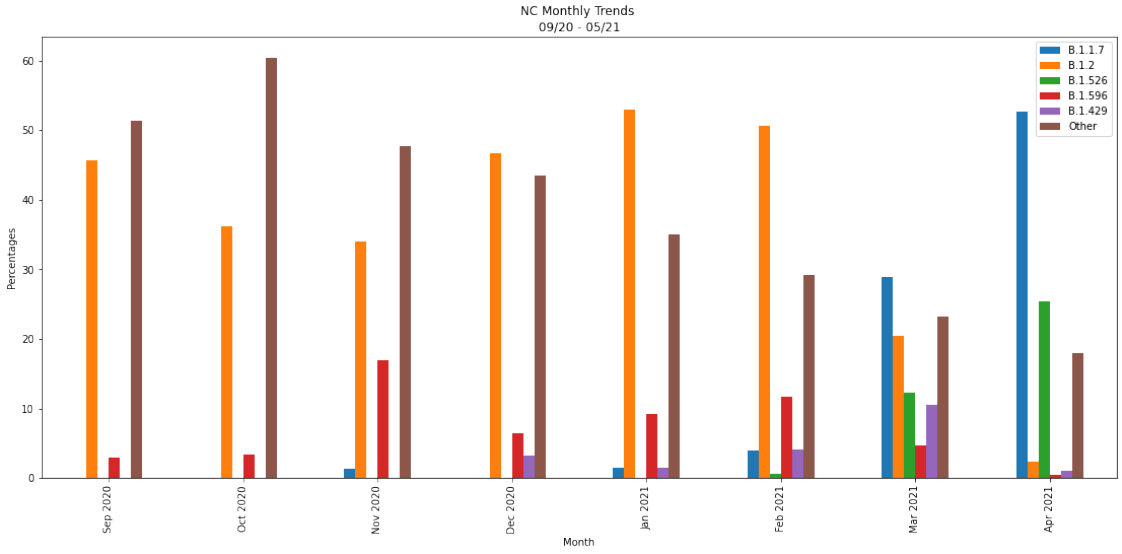
[5]:
```
#Function that gaphs TRENDS (Single F)
def graph_monthly_trends(file_name):
    import pandas as pd
    graph_name=input("Where are these trends from?: ")
    data=pd.read_csv(file_name, sep=",", header=1)

    #Drop count column and round decimal points.
    final_df=data.drop(['Count'], axis=1).round(decimals=2).set_index("Month")
    #test=sc.rename(columns={'# List saved by vdb on 2021-09-08 from command c␣
 ↪= list lineages b': 'SC'})
    print("\n\t\t\tData Frame Used\n",final_df)

    if graph_name=="":
        print(final_df.plot(kind="bar",title='Monthly Trends \n09/20 - 05/
 ↪21',ylabel="Percentages",figsize=(19, 8)))
    else:
        print(final_df.plot(kind="bar",title= graph_name+' Monthly Trends \n09/
 ↪20 - 05/21',ylabel="Percentages",figsize=(19, 8)))

#Run this line to test Function above.
graph_monthly_trends("03_trends_nc.csv")
```

```
Where are these trends from?: NC
```

Data Frame Used

Month	B.1.1.7	B.1.2	B.1.526	B.1.596	B.1.429	Other
Sep 2020	0.00	45.71	0.00	2.86	0.00	51.43
Oct 2020	0.00	36.26	0.00	3.30	0.00	60.44
Nov 2020	1.31	33.99	0.00	16.99	0.00	47.71
Dec 2020	0.00	46.77	0.00	6.45	3.23	43.55
Jan 2021	1.42	52.91	0.00	9.18	1.42	35.06
Feb 2021	3.90	50.62	0.58	11.62	4.07	29.21
Mar 2021	28.83	20.45	12.31	4.64	10.52	23.25
Apr 2021	52.68	2.39	25.43	0.45	1.05	18.00

AxesSubplot(0.125,0.125;0.775x0.755)



SC Trends from 09/01/20 - 05/15/21

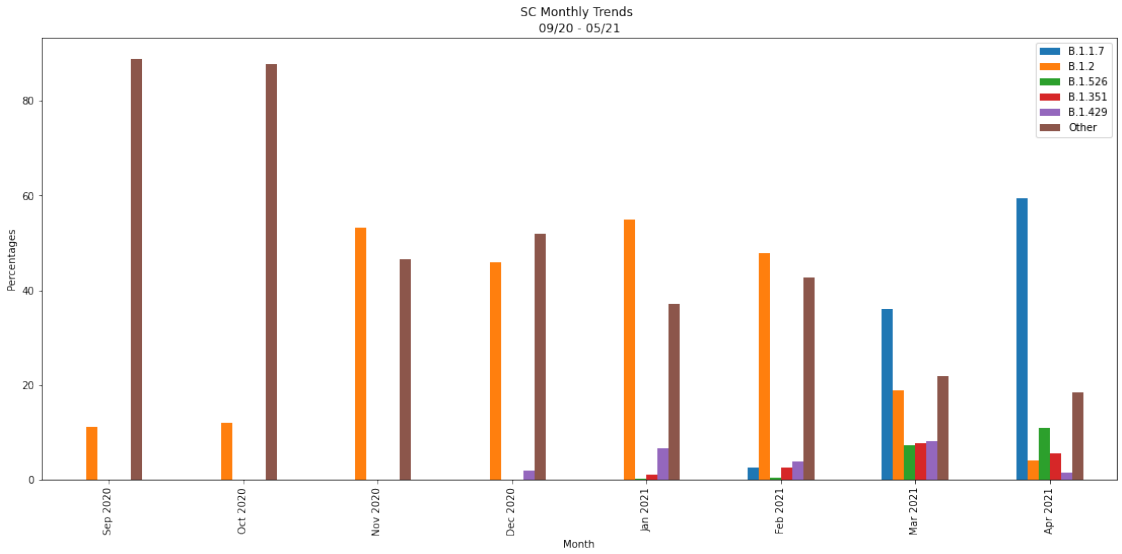
[6]:
```
graph_monthly_trends("03_trends_sc.csv")
```

Where are these trends from?: SC

Data Frame Used

Month	B.1.1.7	B.1.2	B.1.526	B.1.351	B.1.429	Other
Sep 2020	0.00	11.11	0.00	0.00	0.00	88.89
Oct 2020	0.00	12.12	0.00	0.00	0.00	87.88
Nov 2020	0.00	53.33	0.00	0.00	0.00	46.67
Dec 2020	0.00	46.00	0.00	0.00	2.00	52.00
Jan 2021	0.00	54.90	0.23	1.14	6.61	37.13
Feb 2021	2.52	47.83	0.46	2.52	3.89	42.79
Mar 2021	36.02	18.96	7.35	7.82	8.06	21.80
Apr 2021	59.45	4.19	10.89	5.50	1.56	18.42

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



UNCC Trends from 09/01/20 - 05/15/21

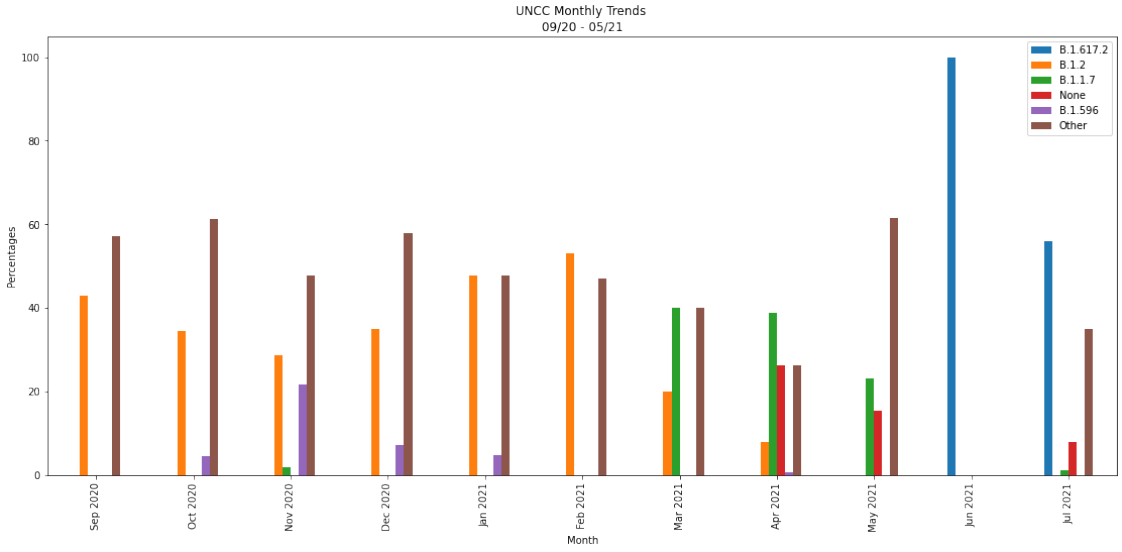
[7]:
```
graph_monthly_trends("03_trends_UNCC.csv")
```

```
Where are these trends from?: UNCC
```

Data Frame Used

Month	B.1.617.2	B.1.2	B.1.1.7	None	B.1.596	Other
Sep 2020	0.00	42.86	0.00	0.00	0.00	57.14
Oct 2020	0.00	34.33	0.00	0.00	4.48	61.19
Nov 2020	0.00	28.70	1.74	0.00	21.74	47.83
Dec 2020	0.00	34.94	0.00	0.00	7.23	57.83
Jan 2021	0.00	47.62	0.00	0.00	4.76	47.62
Feb 2021	0.00	52.94	0.00	0.00	0.00	47.06
Mar 2021	0.00	20.00	40.00	0.00	0.00	40.00
Apr 2021	0.00	7.89	38.82	26.32	0.66	26.32
May 2021	0.00	0.00	23.08	15.38	0.00	61.54
Jun 2021	100.00	0.00	0.00	0.00	0.00	0.00
Jul 2021	56.02	0.00	1.24	7.88	0.00	34.85

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

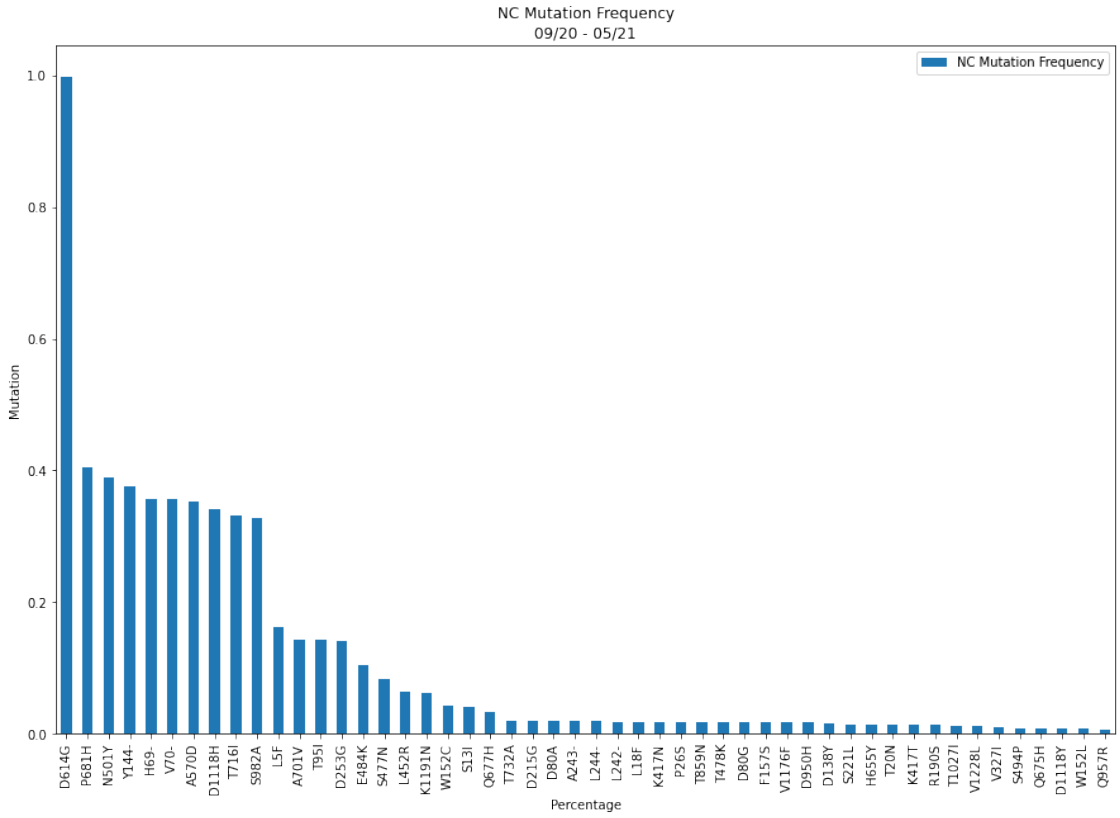


1.3 Mutation Frequency

Lists the frequencies of individual mutations among the viruses belonging to the specified cluster.
NC Mutation Frequency from 09/01/20 - 05/15/21

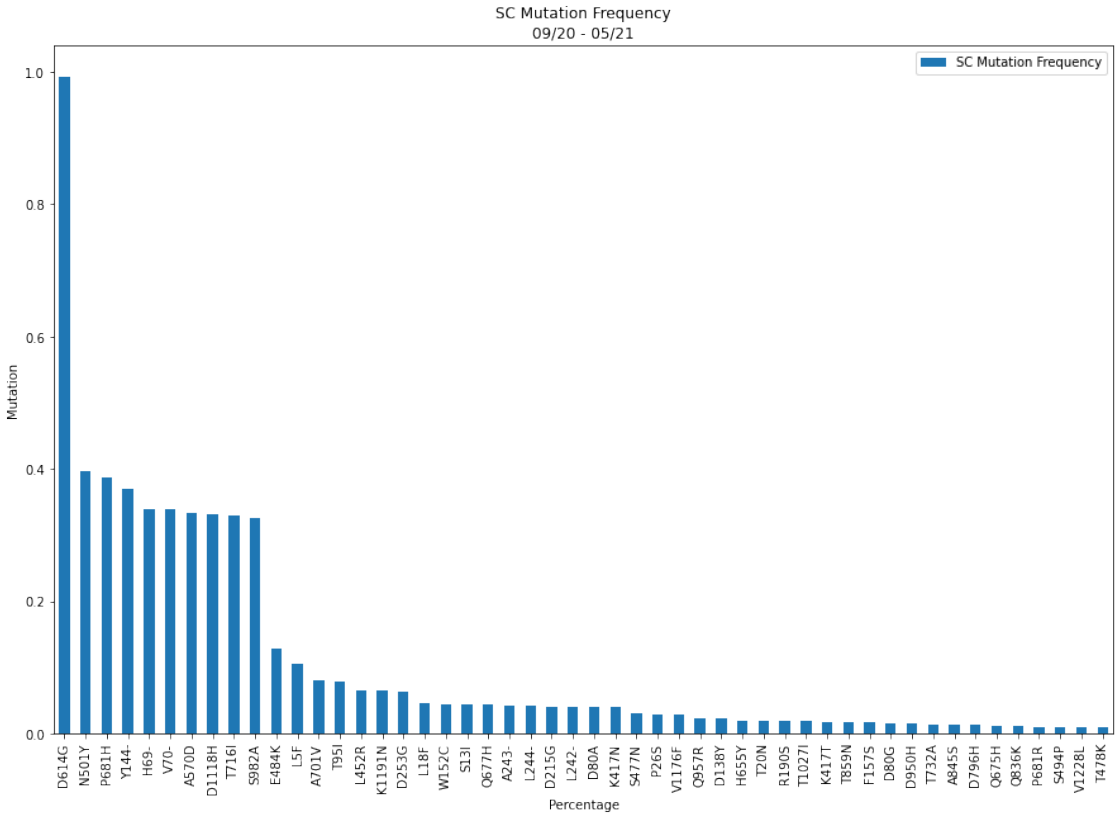
[8]:
```
def mutation_freq(a_file):
    #Mutation frequency
    import pandas as pd
    graph_header=input("Where is this mutation frequency from? (i.e Country or␣
 ↪State): " )
    mutation=pd.read_csv(a_file,sep=",",header=0)
    final_mutation=mutation.rename(columns={get_name(a_file): graph_header+"␣
 ↪Mutation Frequency"})

    #Uncomment to print dataframe
    #print(final_mutation)


    final_mutation.plot(title=graph_header+" Mutation Frequency\n09/20 - 05/
 ↪21",ylabel="Mutation",xlabel="Percentage",kind="bar",figsize=(15, 10))

#Uncomment to test function.
mutation_freq("04_mut_freq_nc.csv")
```

```
Where is this mutation frequency from? (i.e Country or State): NC
```



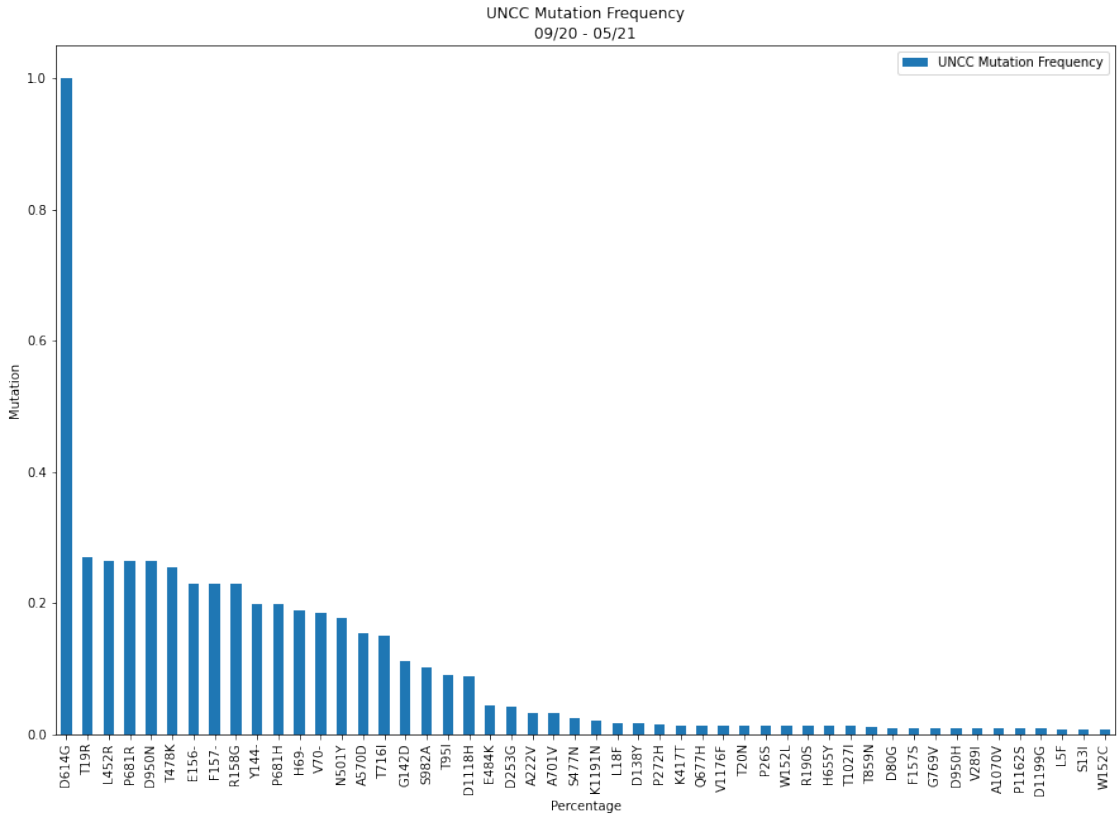
[9]:
```
mutation_freq("04_mut_freq_sc.csv")
```

```
Where is this mutation frequency from? (i.e Country or State): SC
```



[10]:
```
mutation_freq("04_mut_freq_uncc.csv")
```

```
Where is this mutation frequency from? (i.e Country or State): UNCC
```


UNCC Mutation Frequency
09/20 - 05/21
UNCC Mutation Frequency
Mutation
Percentage
1.0
0.8
0.6
0.4
0.2
0.0
D614G
T19R
L452R
P681R
D950N
T478K
E156-
F157-
R158G
Y144-
P681H
H69-
V70-
N501Y
A570D
T716I
G142D
S982A
T95I
D1118H
E484K
D253G
A222V
A701V
S477N
K1191N
L18F
D138Y
P272H
K417T
Q677H
V1176F
T20N
P26S
W152L
R190S
H655Y
T1027I
T859N
D80G
F157S
G769V
D950H
V289I
A1070V
P1162S
D1199G
L5F
S13I
W152C

[ ]: