Exploratory Data Analysis of COVID-19 Clinical Trials

November 11, 2024

1 About Me

Mohammed Imam Uddin Riad
Data Analyst

Connect With Me - Portfolio - LinkedIn - Email Me

Email Me

2 Project:Exploratory Data Analysis of COVID-19 Clinical Trials

In this project, titled "Exploratory Data Analysis of COVID-19 Clinical Trials," I performed a comprehensive analysis of clinical trial data related to COVID-19. Here's a summary of my approach and the key tasks I accomplished:

1. **Data Import and Initial Assessment**:
   - I started by importing the dataset containing 5,783 entries across 27 columns. Initial inspections revealed a diverse range of columns with varying data types and numerous missing values.
2. **Data Cleaning**:
   - To streamline the dataset, I identified and removed columns with high proportions of missing values, such as "Results First Posted" and "Study Documents."
   - I addressed missing values in remaining columns by either filling them with 'unknown' (for categorical data) or performing conversions to ensure consistency across the dataset.
3. **Data Transformation**:
   - I converted key date columns, like "Start Date," "Completion Date," and "First Posted," to datetime format, making it easier to analyze time-based trends.
   - I standardized entries in text columns to ensure consistency. For instance, I unified entries related to COVID-19 in columns like "Acronym" and "Conditions" by replacing all variations with "Covid."
4. **Exploratory Data Analysis (EDA)**:
   - **Univariate Analysis**: I analyzed individual features, such as the "Status" and "Phases" of clinical trials, to understand their distribution. I visualized these distributions to highlight the most common trial statuses (e.g., "Recruiting") and phases (e.g., "Phase 2").
   - **Categorical Analysis**: I examined categorical variables like "Study Type" and "Gender" to explore the characteristics of the trials and visualize them with bar charts.
   - **Conditions and Outcome Measures**: I cross-tabulated conditions with outcome measures to gain insights into the focus of different clinical trials. This provided an

understanding of the most frequently studied COVID-19-related conditions.

5. **Time-Based Analysis**:
   - I explored the "Start Date" column to analyze trends in trial initiation over time. Monthly and yearly breakdowns were generated, revealing how trial frequency changed as the pandemic evolved.
6. **Visualization**:
   - I used various visualization techniques, including bar and line charts, to display trends in trial status, phases, and study types, as well as the frequency of specific conditions in the trials.

Through this project, I showcased my skills in data cleaning, transformation, and visualization, generating valuable insights into the landscape of COVID-19 clinical trials. This analysis highlights the trends, focal points, and status of ongoing research efforts during the pandemic.

[45]:
```
import pandas as pd
```

[46]:
```
raw_data = pd.read_csv("COVID clinical trials.csv")
df = raw_data.copy()
df.head(2)
```

[46]:

	Rank	NCT Number	Title	Acronym	Status	Study Results	Conditions	Interventions	Outcome Measures	Sponsor/Collaborators	…	Other IDs	Start Date	Primary Completion Date	Completion Date	First Posted	Results First Posted	Last Update Posted
0	1	NCT04785898	Diagnostic Performance of the ID Now COVID-19…	COVID-IDNow	Active, not recruiting	No Results Available	Covid19	Diagnostic Test: ID Now	COVID-19 Screening Test	Evaluate the diagnostic performance of the ID …	Groupe Hospitalier Paris Saint Joseph	…	COVID-IDNow	November 9, 2020	December 22, 2020	April 30, 2021	March 8, 2021	NaN	March 8, 2021
1	2	NCT04595136	Study to Evaluate the Efficacy of COVID19-0001…	COVID-19	Not yet recruiting	No Results Available	SARS-CoV-2 Infection	Drug: Drug COVID19-0001-USR|Drug: normal saline	Change on viral load results from baseline aft…	United Medical Specialties	…	COVID19-0001-USR	November 2, 2020	December 15, 2020	January 29, 2021	October 20, 2020	NaN	October 20, 2020

```
                                              Locations Study Documents  \
0  Groupe Hospitalier Paris Saint-Joseph, Paris, …              NaN
1       Cimedical, Barranquilla, Atlantico, Colombia              NaN

                                              URL
0  https://ClinicalTrials.gov/show/NCT04785898
1  https://ClinicalTrials.gov/show/NCT04595136

[2 rows x 27 columns]
```

3 Basic Info

[47]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5783 entries, 0 to 5782
Data columns (total 27 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Rank                     5783 non-null   int64
 1   NCT Number               5783 non-null   object
 2   Title                    5783 non-null   object
 3   Acronym                  2480 non-null   object
 4   Status                   5783 non-null   object
 5   Study Results            5783 non-null   object
 6   Conditions               5783 non-null   object
 7   Interventions            4897 non-null   object
 8   Outcome Measures         5748 non-null   object
 9   Sponsor/Collaborators    5783 non-null   object
 10  Gender                   5773 non-null   object
 11  Age                      5783 non-null   object
 12  Phases                   3322 non-null   object
 13  Enrollment               5749 non-null   float64
 14  Funded Bys               5783 non-null   object
 15  Study Type               5783 non-null   object
 16  Study Designs            5748 non-null   object
 17  Other IDs                5782 non-null   object
 18  Start Date               5749 non-null   object
 19  Primary Completion Date  5747 non-null   object
 20  Completion Date          5747 non-null   object
 21  First Posted             5783 non-null   object
 22  Results First Posted     36 non-null     object
 23  Last Update Posted       5783 non-null   object
 24  Locations                5198 non-null   object
 25  Study Documents          182 non-null    object
 26  URL                      5783 non-null   object
dtypes: float64(1), int64(1), object(25)
```

```
memory usage: 1.2+ MB
```

[48]:
```
df.shape
```

[48]:
```
(5783, 27)
```

[49]:
```
#Summary Statistics for Categorical Columns
df.describe(include='object').T.sort_values(by = ['count','unique'],ascending =␣
↪True)
```

[49]:

	count	unique	top
Results First Posted	36	33	November 4, 2020
Study Documents	182	182	"Statistical Analysis Plan", https://ClinicalT…
Acronym	2480	2338	COVID-19
Phases	3322	8	Not Applicable
Interventions	4897	4337	Other: No intervention
Locations	5198	4255	Uhmontpellier, Montpellier, France
Primary Completion Date	5747	877	December 31, 2020
Completion Date	5747	978	December 31, 2021
Study Designs	5748	267	Observational Model: Cohort|Time Perspective: …
Outcome Measures	5748	5687	Mortality
Start Date	5749	654	
Gender	5773	3	
Other IDs	5782	5734	
Study Results	5783	2	
Study Type	5783	9	
Status	5783	12	
Funded Bys	5783	26	
Last Update Posted	5783	269	
Age	5783	417	
First Posted	5783	438	
Conditions	5783	3067	
Sponsor/Collaborators	5783	3631	
Title	5783	5775	
NCT Number	5783	5783	
URL	5783	5783	

Start Date	May 1, 2020
Gender	All
Other IDs	COVID-19
Study Results	No Results Available
Study Type	Interventional
Status	Recruiting
Funded Bys	Other
Last Update Posted	April 8, 2021
Age	18 Years and older (Adult, Older Adult)
First Posted	April 24, 2020
Conditions	COVID-19
Sponsor/Collaborators	Assistance Publique - Hôpitaux de Paris
Title	Study Assessing Vagus Nerve Stimulation in CoV…
NCT Number	NCT04785898
URL	https://ClinicalTrials.gov/show/NCT04785898

	freq
Results First Posted	2
Study Documents	1
Acronym	47
Phases	1354
Interventions	32
Locations	19
Primary Completion Date	122
Completion Date	179
Study Designs	1011
Outcome Measures	5
Start Date	113
Gender	5567
Other IDs	6
Study Results	5747
Study Type	3322
Status	2805
Funded Bys	4488
Last Update Posted	109
Age	2885
First Posted	108
Conditions	720
Sponsor/Collaborators	78
Title	2
NCT Number	1
URL	1

4 Dropping Unnecessary Columns

[50]:
```
#dropping columns as they have enourmous amount of missing values
col_to_drop = ['Results First Posted','Study Documents','Study Results','Rank']
for i in col_to_drop:
    df.drop([i],axis =1,inplace = True)
```

[51]:
```
df = df.drop(['URL'],axis = 1)
df.columns
```

[51]:
```
Index(['NCT Number', 'Title', 'Acronym', 'Status', 'Conditions',
       'Interventions', 'Outcome Measures', 'Sponsor/Collaborators', 'Gender',
       'Age', 'Phases', 'Enrollment', 'Funded Bys', 'Study Type',
       'Study Designs', 'Other IDs', 'Start Date', 'Primary Completion Date',
       'Completion Date', 'First Posted', 'Last Update Posted', 'Locations'],
      dtype='object')
```

[52]:
```
df.shape
```

[52]:
```
(5783, 22)
```

5 Duplicted Rows

[53]:
```
df.duplicated().any()
```

[53]:
```
False
```

6 Handling Missing Values

[54]:
```
#Displaying Columns with Missing Values in Descending Order

df.isnull().sum()[df.isnull().sum().sort_values(ascending = False)>0].
 ↪sort_values(ascending = False)
```

[54]:
```
Acronym                    3303
Phases                     2461
Interventions               886
Locations                   585
Primary Completion Date      36
Completion Date              36
Outcome Measures             35
Study Designs                35
Enrollment                   34
Start Date                   34
Gender                       10
Other IDs                     1
```

```
dtype: int64
```

[55]:
```
#filling the missing rows with 'unknown'. But before that convert the whole␣
 ↪data set into object

# Convert the entire DataFrame to object type
df = df.astype(object)

# Now fill missing values with 'unknown'
df.fillna('unknown', inplace=True)

#checking is there any missing values now
df.isnull().sum()
```

[55]:
```
NCT Number                0
Title                     0
Acronym                   0
Status                    0
Conditions                0
Interventions             0
Outcome Measures          0
Sponsor/Collaborators     0
Gender                    0
Age                       0
Phases                    0
Enrollment                0
Funded Bys                0
Study Type                0
Study Designs             0
Other IDs                 0
Start Date                0
Primary Completion Date   0
Completion Date           0
First Posted              0
Last Update Posted        0
Locations                 0
dtype: int64
```

7 Creating a Checkpoint

[56]:
```
df_cleaned = df.copy()
df_cleaned.head(3)
```

[56]:
```
     NCT Number                                              Title  \
0  NCT04785898  Diagnostic Performance of the ID Now  COVID-19…
1  NCT04595136  Study to Evaluate the Efficacy of COVID19-0001…
```

```
2  NCT04395482  Lung CT Scan Analysis of SARS-CoV2 Induced Lun…

       Acronym                   Status            Conditions  \
0  COVID-IDNow  Active, not recruiting               Covid19
1     COVID-19      Not yet recruiting  SARS-CoV-2 Infection
2  TAC-COVID19              Recruiting               covid19

                                      Interventions  \
0   Diagnostic Test: ID Now  COVID-19 Screening Test
1    Drug: Drug COVID19-0001-USR|Drug: normal saline
2  Other: Lung CT scan analysis in COVID-19 patients

                                    Outcome Measures  \
0  Evaluate the diagnostic performance of the ID …
1  Change on viral load results from baseline aft…
2  A qualitative analysis of parenchymal lung dam…

                   Sponsor/Collaborators Gender  \
0  Groupe Hospitalier Paris Saint Joseph    All
1              United Medical Specialties    All
2            University of Milano Bicocca    All

                                        Age  … Funded Bys      Study Type  \
0  18 Years and older   (Adult, Older Adult)  …      Other  Interventional
1  18 Years and older   (Adult, Older Adult)  …      Other  Interventional
2  18 Years and older   (Adult, Older Adult)  …      Other   Observational

                                     Study Designs         Other IDs  \
0  Allocation: N/A|Intervention Model: Single Gro…       COVID-IDNow
1  Allocation: Randomized|Intervention Model: Par…  COVID19-0001-USR
2  Observational Model: Cohort|Time Perspective: …       TAC-COVID19

         Start Date Primary Completion Date   Completion Date  \
0  November 9, 2020       December 22, 2020    April 30, 2021
1  November 2, 2020       December 15, 2020  January 29, 2021
2       May 7, 2020           June 15, 2021     June 15, 2021

       First Posted Last Update Posted  \
0    March 8, 2021      March 8, 2021
1  October 20, 2020   October 20, 2020
2     May 20, 2020   November 9, 2020

                                          Locations
0  Groupe Hospitalier Paris Saint-Joseph, Paris, …
1       Cimedical, Barranquilla, Atlantico, Colombia
2  Ospedale Papa Giovanni XXIII, Bergamo, Italy|P…
```

```
[3 rows x 22 columns]
```

[57]:
```
#saving the cleaned data
df_cleaned.to_csv('Cleaned_COVID-19 Clinical Trials.csv', index=False)
```

8 Data Manipulation

[58]:
```
# Show all columns and rows
pd.options.display.max_columns = None
pd.options.display.max_rows = None

# Reset all display options to default
pd.reset_option('display')

import numpy as np
np.set_printoptions(threshold=np.inf)
```

8.0.1 Acronym column

[59]:
```
df_cleaned['Acronym'].unique().size
```

[59]:
```
2339
```

8.0.2 Standardize Acronym Column by Replacing Entries that are Containing 'covid' with 'Covid

[60]:
```
df_cleaned['Acronym'] = df_cleaned['Acronym'].apply(lambda x: 'Covid' if␣
 ↪'covid' in str(x).lower() else x)
df_cleaned['Acronym'].unique().size
```

[60]:
```
1650
```

All the values that contain string "covid" now is being displayed as "Covid".So the unique values are 1650 now

[61]:
```
df_cleaned['Status'].unique()
```

[61]:
```
array(['Active, not recruiting', 'Not yet recruiting', 'Recruiting',
       'Enrolling by invitation', 'Suspended', 'Completed', 'Withdrawn',
       'Terminated', 'No longer available', 'Available',
       'Approved for marketing', 'Temporarily not available'],
      dtype=object)
```

8.0.3 Conditions Column

[62]:
```
df_cleaned['Conditions'].unique().size
```

[62]: 3067

[63]:
```
df_cleaned['Conditions'] = df_cleaned['Conditions'].apply(lambda x: 'Covid' if␣
↪'covid' in str(x).lower() else x)
df_cleaned['Conditions'].unique().size
```

[63]: 1339

[64]:
```
df_cleaned['Outcome Measures'].unique().size
```

[64]: 5688

[65]:
```
df_cleaned['Sponsor/Collaborators'].unique().size
```

[65]: 3631

8.0.4 Gender Columns

[66]:
```
df_cleaned['Gender'] = df_cleaned['Gender'].replace({'All': 'Male','unknown' :␣
↪'other'})
df_cleaned['Gender'].unique()
```

[66]: array(['Male', 'Female', 'other'], dtype=object)

[67]:
```
df_cleaned['Age'].value_counts().sort_values(ascending = False).head()
```

[67]:
```
Age
18 Years and older   (Adult, Older Adult)      2885
Child, Adult, Older Adult                       486
18 Years to 80 Years   (Adult, Older Adult)     221
18 Years to 65 Years   (Adult, Older Adult)     155
18 Years to 75 Years   (Adult, Older Adult)     135
Name: count, dtype: int64
```

[68]:
```
df_cleaned['Phases'].unique()
```

[68]:
```
array(['Not Applicable', 'Phase 1|Phase 2', 'unknown', 'Early Phase 1',
       'Phase 2|Phase 3', 'Phase 1', 'Phase 4', 'Phase 2', 'Phase 3'],
      dtype=object)
```

[69]:
```
df_cleaned['Enrollment'].unique().size
```

[69]: 963

[70]:
```
df_cleaned['Funded Bys'].unique()
```

[70]:
```
array(['Other', 'Industry', 'Industry|Other', 'Other|Industry',
       'Other|U.S. Fed', 'NIH', 'Other|NIH', 'NIH|Other|Industry',
```

```
       'NIH|Other', 'NIH|Industry', 'Industry|U.S. Fed', 'U.S. Fed|Other',
       'Other|U.S. Fed|NIH', 'Industry|U.S. Fed|Other',
       'Other|NIH|U.S. Fed', 'Industry|NIH|Other',
       'Industry|Other|U.S. Fed', 'Industry|NIH',
       'Other|U.S. Fed|Industry', 'U.S. Fed', 'NIH|Industry|Other',
       'NIH|Other|U.S. Fed|Industry', 'Industry|U.S. Fed|NIH',
       'Other|Industry|NIH', 'Other|NIH|Industry', 'Industry|Other|NIH'],
      dtype=object)
```

[71]:
```
df_cleaned['Study Type'].value_counts()
```

[71]:
```
Study Type
Interventional                                                  3322
Observational                                                   2427
Expanded Access:Intermediate-size Population                      15
Expanded Access:Treatment IND/Protocol                             8
Expanded Access:Intermediate-size Population|Treatment IND/Protocol    5
Expanded Access:Individual Patients                                3
Expanded Access:Individual Patients|Intermediate-size Population    1
Expanded Access                                                    1
Expanded Access:Individual Patients|Treatment IND/Protocol         1
Name: count, dtype: int64
```

[72]:
```
df_cleaned['Study Designs'].unique().size
```

[72]:
```
268
```

[73]:
```
df_cleaned['Other IDs'].unique().size
```

[73]:
```
5735
```

8.0.5 Start Date columns

[74]:
```
df_cleaned['Start Date'].head()
```

[74]:
```
0    November 9, 2020
1    November 2, 2020
2         May 7, 2020
3        May 25, 2020
4         May 5, 2020
Name: Start Date, dtype: object
```

[75]:
```
#changing datatypes to datetime

# Function to handle both formats
def parse_dates(date_str):
    try:
        # Try parsing the full date (e.g., 'March 23, 2020')
```

```
        return pd.to_datetime(date_str, errors='coerce', dayfirst=False)
    except:
        # If that fails, try parsing the month-year format (e.g., 'April 2021')
        try:
            return pd.to_datetime(date_str + ' 01', errors='coerce')  # Adding␣
 ↪'01' as the first day of the month
        except:
            return pd.NaT  # If both fail, return NaT

# Apply the function to the 'Start Date' column
df_cleaned['Start Date']= df_cleaned['Start Date'].apply(parse_dates)

# Check the result
print(df_cleaned['Start Date'].head())
```

```
0   2020-11-09
1   2020-11-02
2   2020-05-07
3   2020-05-25
4   2020-05-05
Name: Start Date, dtype: datetime64[ns]
```

8.0.6 Primary Completion Date columns

[76]:
```
#changing datatypes to datetime

def parse_dates(date_str):
    try:
        return pd.to_datetime(date_str,errors = 'coerce', dayfirst = False)
    except:
        try:
            return pd.to_datetime(date_str + '01' ,errors = 'coerce')
        except:
            return pd.NaT

df_cleaned['Primary Completion Date'] = df_cleaned['Primary Completion Date'].
 ↪apply(parse_dates)
df_cleaned['Primary Completion Date'].head()
```

[76]:
```
0   2020-12-22
1   2020-12-15
2   2021-06-15
3   2020-07-31
4   2021-05-01
Name: Primary Completion Date, dtype: datetime64[ns]
```

8.0.7 Completion Date column

[77]:
```
#changing datatypes to datetime

df_cleaned['Completion Date'] = df_cleaned['Completion Date'].apply(parse_dates)
df_cleaned['Completion Date'].head()
```

[77]:
```
0   2021-04-30
1   2021-01-29
2   2021-06-15
3   2020-08-31
4   2021-05-01
Name: Completion Date, dtype: datetime64[ns]
```

8.0.8 First Posted Column

[78]:
```
df_cleaned['First Posted'] = df_cleaned['First Posted'].apply(parse_dates)
df_cleaned['First Posted'].head(5)
```

[78]:
```
0   2021-03-08
1   2020-10-20
2   2020-05-20
3   2020-06-04
4   2020-05-20
Name: First Posted, dtype: datetime64[ns]
```

8.0.9 Last Update Posted column

[79]:
```
df_cleaned['Last Update Posted'] =df_cleaned['Last Update Posted'].
  ↪apply(parse_dates)
df_cleaned['Last Update Posted'] .head()
```

[79]:
```
0   2021-03-08
1   2020-10-20
2   2020-11-09
3   2020-06-04
4   2020-06-04
Name: Last Update Posted, dtype: datetime64[ns]
```

[80]:
```
df_cleaned['Locations'].unique().size
```

[80]:
```
4256
```

[81]:
```
df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5783 entries, 0 to 5782
Data columns (total 22 columns):
 #   Column                    Non-Null Count  Dtype
```

```
---  ------                    --------------  -----
 0   NCT Number                5783 non-null   object
 1   Title                     5783 non-null   object
 2   Acronym                   5783 non-null   object
 3   Status                    5783 non-null   object
 4   Conditions                5783 non-null   object
 5   Interventions             5783 non-null   object
 6   Outcome Measures          5783 non-null   object
 7   Sponsor/Collaborators     5783 non-null   object
 8   Gender                    5783 non-null   object
 9   Age                       5783 non-null   object
 10  Phases                    5783 non-null   object
 11  Enrollment                5783 non-null   object
 12  Funded Bys                5783 non-null   object
 13  Study Type                5783 non-null   object
 14  Study Designs             5783 non-null   object
 15  Other IDs                 5783 non-null   object
 16  Start Date                5749 non-null   datetime64[ns]
 17  Primary Completion Date   5747 non-null   datetime64[ns]
 18  Completion Date           5747 non-null   datetime64[ns]
 19  First Posted              5783 non-null   datetime64[ns]
 20  Last Update Posted        5783 non-null   datetime64[ns]
 21  Locations                 5783 non-null   object
dtypes: datetime64[ns](5), object(17)
memory usage: 994.1+ KB
```

9 Checkpoint 2

[82]:
```
df_mod = df_cleaned.copy()
df_mod.head(2)
```

[82]:
```
     NCT Number                                                Title Acronym  \
0  NCT04785898  Diagnostic Performance of the ID Now  COVID-19…   Covid
1  NCT04595136  Study to Evaluate the Efficacy of COVID19-0001…   Covid

                   Status            Conditions  \
0  Active, not recruiting                 Covid
1      Not yet recruiting  SARS-CoV-2 Infection

                                      Interventions  \
0  Diagnostic Test: ID Now  COVID-19 Screening Test
1   Drug: Drug COVID19-0001-USR|Drug: normal saline

                                    Outcome Measures  \
0  Evaluate the diagnostic performance of the ID …
1  Change on viral load results from baseline aft…
```

```
                    Sponsor/Collaborators Gender  \
0  Groupe Hospitalier Paris Saint Joseph   Male
1             United Medical Specialties   Male

                                        Age  … Funded Bys      Study Type  \
0  18 Years and older   (Adult, Older Adult)  …      Other  Interventional
1  18 Years and older   (Adult, Older Adult)  …      Other  Interventional

                                     Study Designs         Other IDs  \
0  Allocation: N/A|Intervention Model: Single Gro…       COVID-IDNow
1  Allocation: Randomized|Intervention Model: Par…  COVID19-0001-USR

  Start Date Primary Completion Date Completion Date First Posted  \
0 2020-11-09              2020-12-22      2021-04-30   2021-03-08
1 2020-11-02              2020-12-15      2021-01-29   2020-10-20

  Last Update Posted                                          Locations
0         2021-03-08  Groupe Hospitalier Paris Saint-Joseph, Paris, …
1         2020-10-20       Cimedical, Barranquilla, Atlantico, Colombia

[2 rows x 22 columns]
```

10 Univariate Analysis

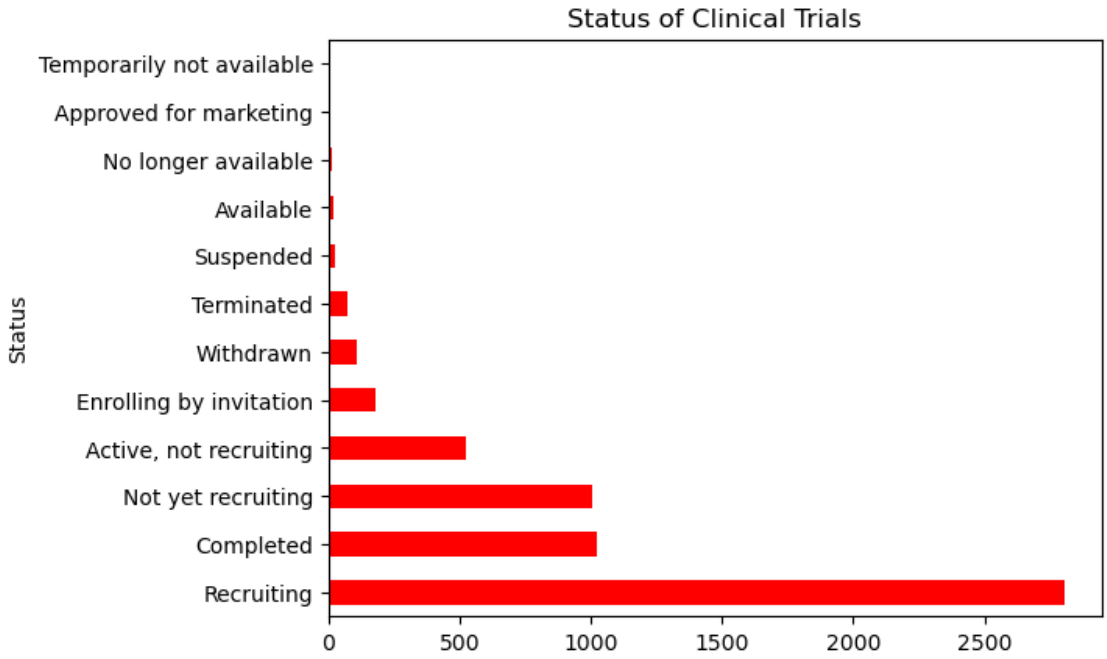
10.0.1 Status Distribution

[83]:
```
df_mod['Status'].value_counts()
```

[83]:
```
Status
Recruiting                  2805
Completed                   1025
Not yet recruiting          1004
Active, not recruiting       526
Enrolling by invitation      181
Withdrawn                    107
Terminated                    74
Suspended                     27
Available                     19
No longer available           12
Approved for marketing         2
Temporarily not available      1
Name: count, dtype: int64
```

[84]:
```
df_mod['Status'].value_counts().plot(kind='barh', title='Status of Clinical␣
↪Trials',color = 'red')
```

[84]:
```
<Axes: title={'center': 'Status of Clinical Trials'}, ylabel='Status'>
```



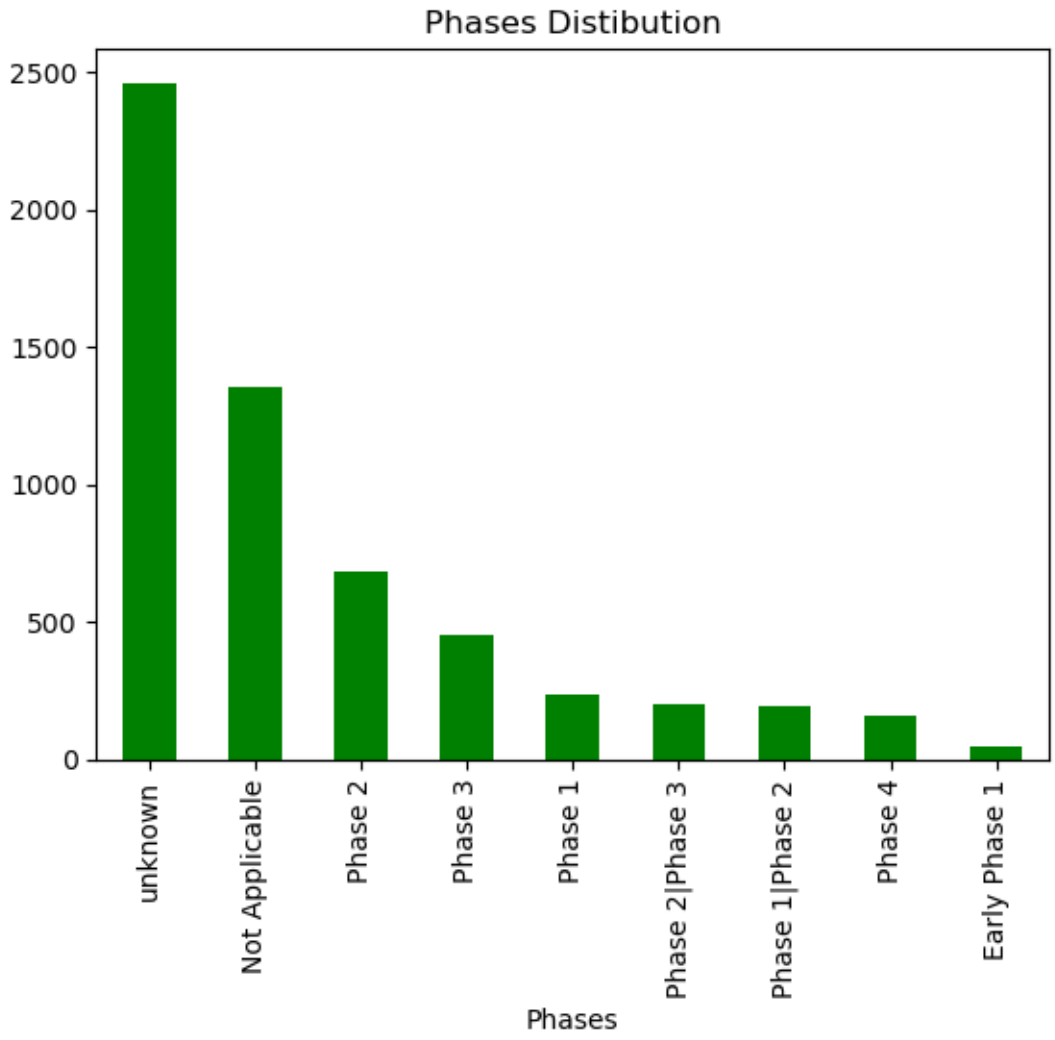
10.0.2 Phases Distibution

[85]:
```
df_mod['Phases'].value_counts()
```

[85]:
```
Phases
unknown             2461
Not Applicable      1354
Phase 2              685
Phase 3              450
Phase 1              234
Phase 2|Phase 3      200
Phase 1|Phase 2      192
Phase 4              161
Early Phase 1         46
Name: count, dtype: int64
```

[86]:
```
df_mod['Phases'].value_counts().plot(kind = 'bar',title = 'Phases␣
↪Distibution',color = 'green')
```

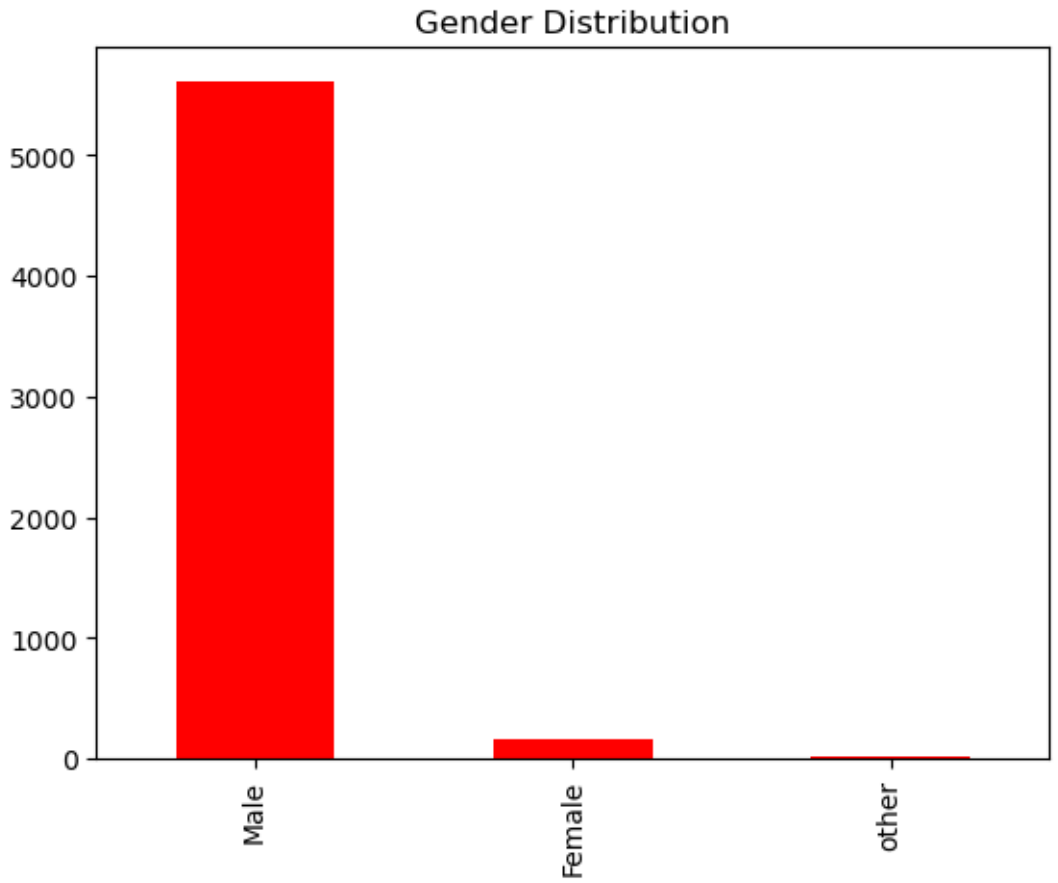
[86]:
```
<Axes: title={'center': 'Phases Distibution'}, xlabel='Phases'>
```



10.0.3 Gender Distibution

[87]:
```
df_mod['Gender'].value_counts().plot(kind = 'bar', title = 'Gender␣
↪Distribution',color = 'red',xlabel = '' ,ylabel = '')
```

[87]: <Axes: title={'center': 'Gender Distribution'}>



10.0.4 Study Distribution

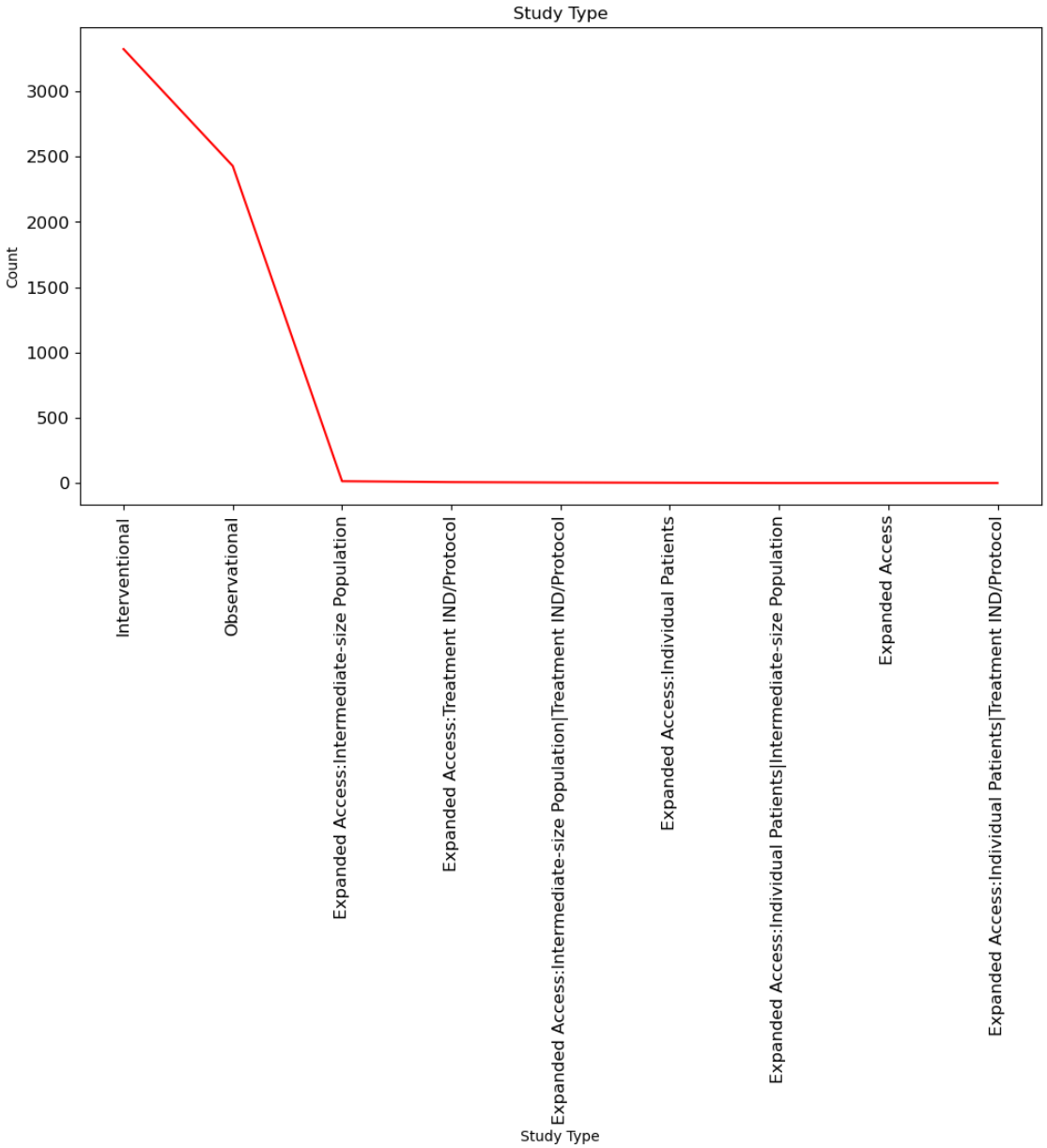
[88]:
```
df['Study Type'].value_counts()[df['Study Type'].value_counts()>=1]
```

[88]:
```
Study Type
Interventional                                                3322
Observational                                                 2427
Expanded Access:Intermediate-size Population                    15
Expanded Access:Treatment IND/Protocol                           8
Expanded Access:Intermediate-size Population|Treatment IND/Protocol    5
Expanded Access:Individual Patients                              3
Expanded Access:Individual Patients|Intermediate-size Population    1
Expanded Access                                                  1
Expanded Access:Individual Patients|Treatment IND/Protocol       1
Name: count, dtype: int64
```

[89]:
```
df['Study Type'].value_counts().plot(
    kind='line',
    title='Study Type',
```

```
    color='red',
    figsize=(12, 6),
    fontsize=12,
    xlabel='Study Type',
    ylabel='Count',
    grid=False,
    rot=90  # Rotate x-axis labels by 45 degrees
)
```

[89]: <Axes: title={'center': 'Study Type'}, xlabel='Study Type', ylabel='Count'>



10.0.5 Conditions vs. Outcome Measures:

[90]:
```
status_phase = pd.crosstab(df_mod['Status'], df_mod['Phases'])
status_phase
```

[90]:

Phases	Early Phase 1	Not Applicable	Phase 1	\
Status				
Active, not recruiting	7	111	44	
Approved for marketing	0	0	0	
Available	0	0	0	
Completed	3	226	38	
Enrolling by invitation	4	54	1	
No longer available	0	0	0	
Not yet recruiting	5	282	42	
Recruiting	22	647	98	
Suspended	2	2	0	
Temporarily not available	0	0	0	
Terminated	0	13	4	
Withdrawn	3	19	7	

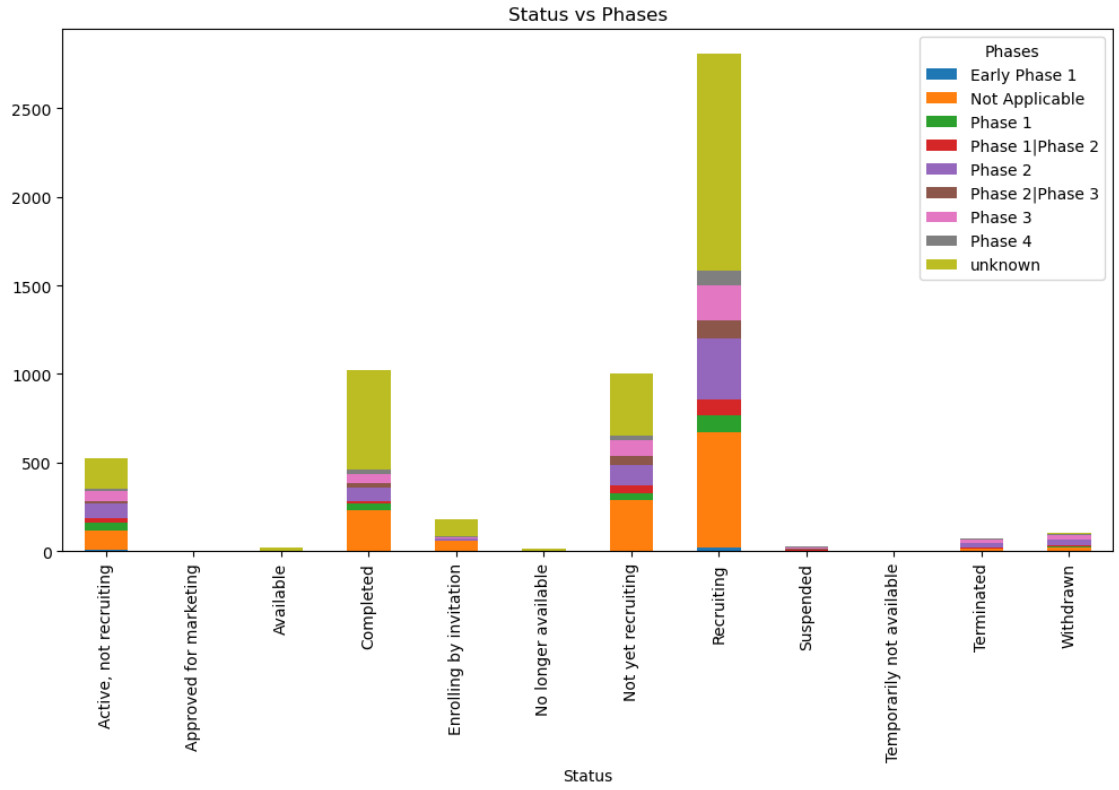
Phases	Phase 1|Phase 2	Phase 2	Phase 2|Phase 3	Phase 3	\
Status					
Active, not recruiting	26	81	15	59	
Approved for marketing	0	0	0	0	
Available	0	0	0	0	
Completed	17	78	20	56	
Enrolling by invitation	3	10	1	6	
No longer available	0	0	0	0	
Not yet recruiting	46	114	46	89	
Recruiting	92	343	102	196	
Suspended	2	4	4	9	
Temporarily not available	0	0	0	0	
Terminated	2	25	6	15	
Withdrawn	4	30	6	20	

Phases	Phase 4	unknown
Status		
Active, not recruiting	8	175
Approved for marketing	0	2
Available	0	19
Completed	22	565
Enrolling by invitation	6	96
No longer available	0	12
Not yet recruiting	30	350
Recruiting	81	1224
Suspended	2	2
Temporarily not available	0	1
Terminated	5	4

```
Withdrawn                          7        11
```

[91]:
```
status_phase.plot(kind='bar', stacked=True, title='Status vs Phases',figsize =
 ↪(12,6),rot = 90)
```

[91]:
```
<Axes: title={'center': 'Status vs Phases'}, xlabel='Status'>
```

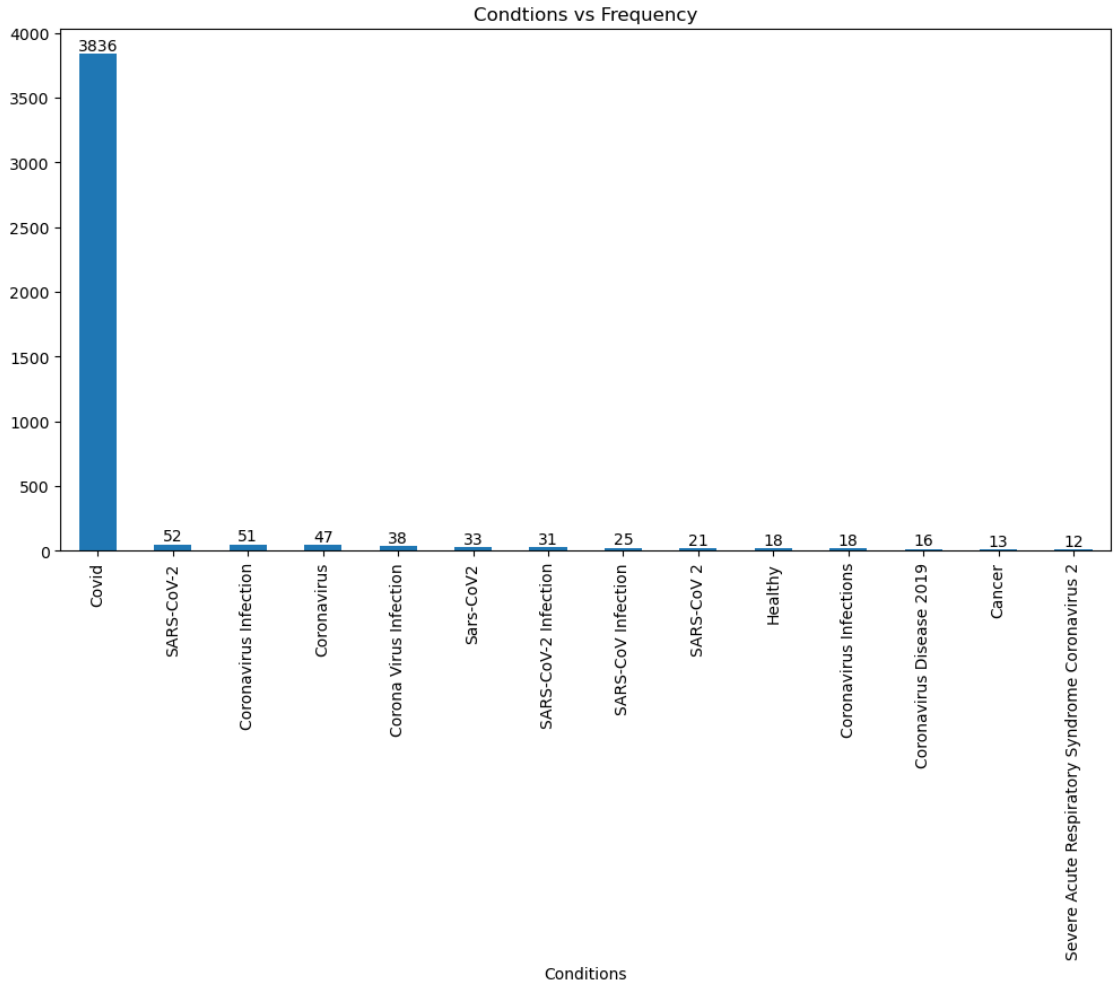


[92]:
```
#conditions that are counted more than 10
conditions = df_mod['Conditions'].value_counts()[df_mod['Conditions'].
 ↪value_counts()>10]
conditions
```

[92]:
```
Conditions
Covid                                3836
SARS-CoV-2                             52
Coronavirus Infection                  51
Coronavirus                            47
Corona Virus Infection                 38
Sars-CoV2                              33
SARS-CoV-2 Infection                   31
SARS-CoV Infection                     25
SARS-CoV 2                             21
```

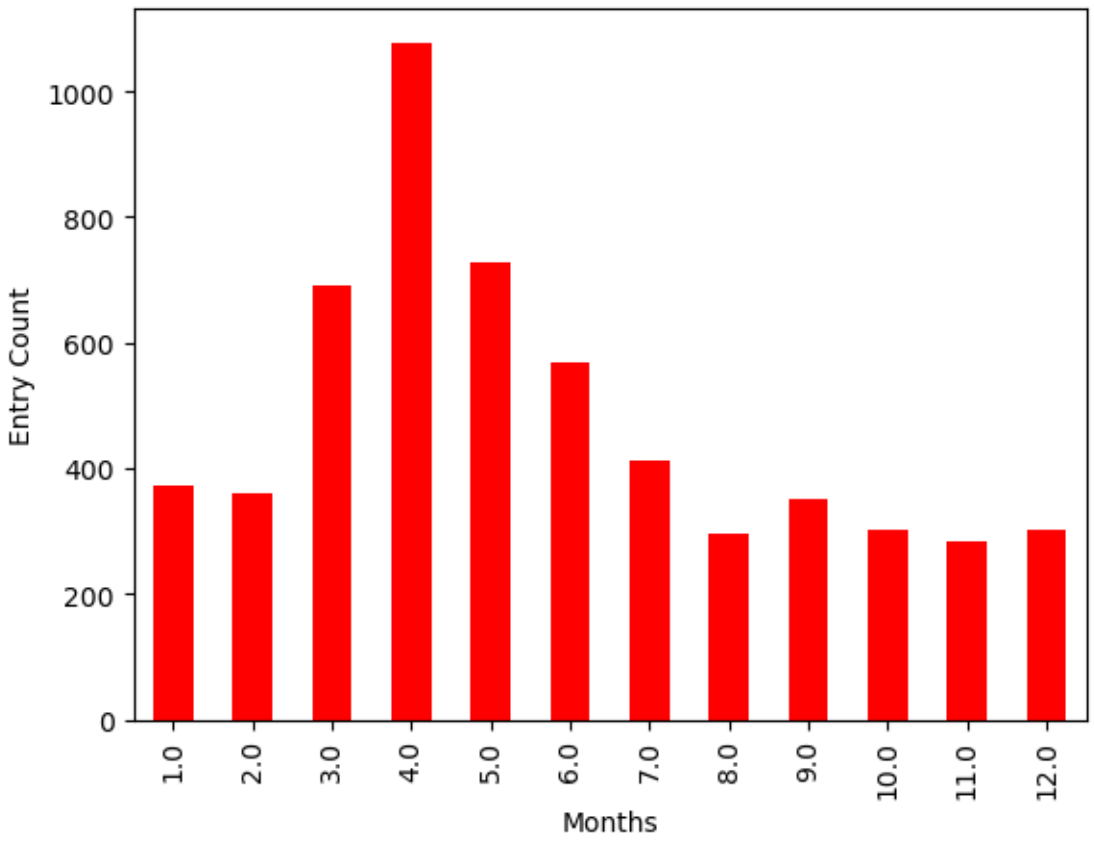
```
Healthy                                                 18
Coronavirus Infections                                  18
Coronavirus Disease 2019                                16
Cancer                                                  13
Severe Acute Respiratory Syndrome Coronavirus 2         12
Name: count, dtype: int64
```

[95]:
```python
import matplotlib.pyplot as plt
ax = conditions.plot(kind = 'bar',title='Condtions vs Frequency',figsize =␣
 ↪(12,6),rot = 90)

# Add the values on top of each bar
for p in ax.patches:
    ax.annotate(
        str(p.get_height()),            # The text to display, here the height␣
 ↪of each bar
        (p.get_x() + p.get_width() / 2, p.get_height()),  # Position (x, y)
        ha='center',                    # Center-align the text horizontally
        va='bottom'                     # Position text just above the bar
    )

# Display the plot
plt.show()
```



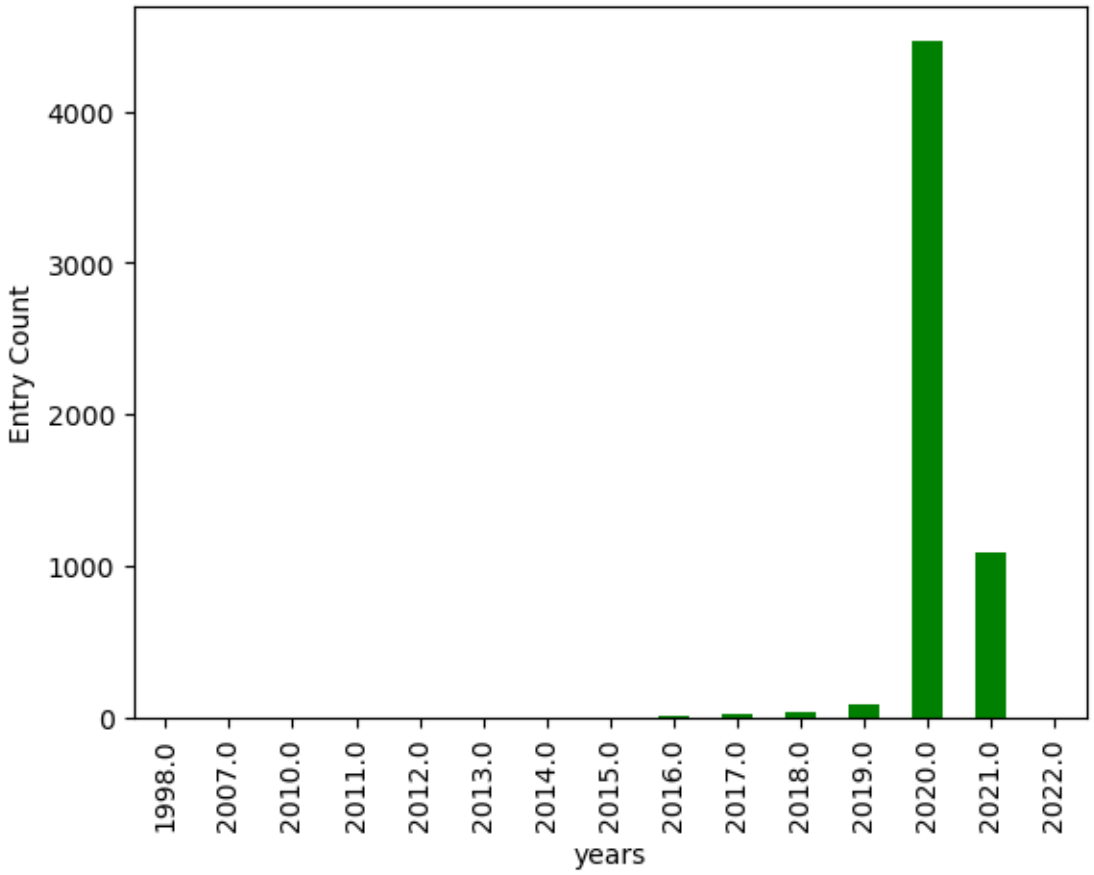
[96]:
```
month = df_mod['Start Date'].dt.month

df_mod.groupby([month])['NCT Number'].count().plot(kind = 'bar',xlabel =␣
↪'Months',ylabel = 'Entry Count',color = 'red')
```

[96]: <Axes: xlabel='Months', ylabel='Entry Count'>



[97]:
```
year = df_mod['Start Date'].dt.year
df_mod.groupby([year])['NCT Number'].count().plot(kind = 'bar',xlabel =␣
↪'years',ylabel = 'Entry Count',color = 'green')
```

[97]:
```
<Axes: xlabel='years', ylabel='Entry Count'>
```


Entry Count
4000
3000
2000
1000
0
1998.0
2007.0
2010.0
2011.0
2012.0
2013.0
2014.0
2015.0
2016.0
2017.0
2018.0
2019.0
2020.0
2021.0
2022.0
years

[ ]: