EDA on COVID-19 Clinical Trials

1. Introduction

The COVID-19 pandemic initiated an unprecedented surge in clinical research activities around the world. Thousands of clinical trials were launched to discover treatments, vaccines, and diagnostic methods.

This project performs **Exploratory Data Analysis (EDA)** on a dataset of **COVID-19 related clinical trials** to understand trends, patterns, and important insights.

2. Objectives

- To explore the **status** of clinical trials.
- To analyze **study types** (Interventional, Observational).
- To identify the most common **medical conditions** studied.
- To find the major **sponsors** behind the studies.
- To study the **geographic distribution** of trials.
- To investigate the **phases** of Interventional studies.

3. Dataset Overview

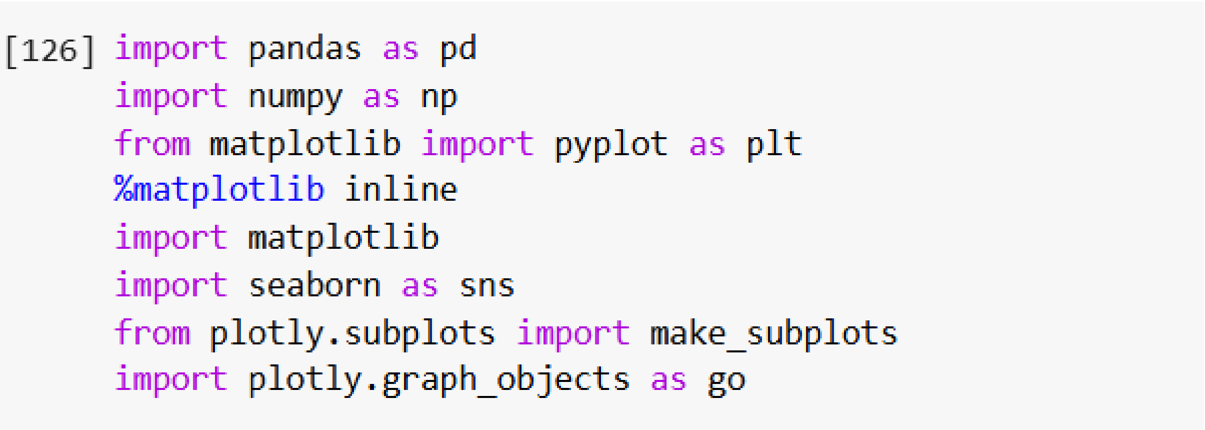
The data is loaded from a CSV file using Python's pandas library.

The dataset contains columns such as:

- **Status**: Current state of the trial (e.g., Completed, Recruiting).
- **Conditions**: Disease(s) being studied.
- **Sponsor**: Organization leading or funding the study.
- **Study Type**: Type of research (Interventional, Observational).
- **Phase**: Phase of the trial (Phase 1–4).
- **Locations**: Where the trial is being conducted

4. Code, Explanation

4.1 Import Visualization Libraries

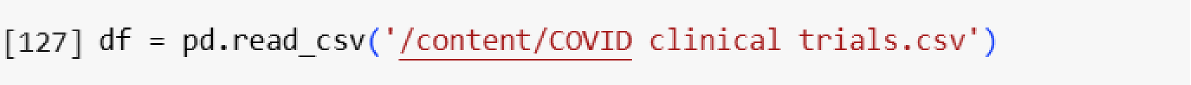


```
[126] import pandas as pd
      import numpy as np
      from matplotlib import pyplot as plt
      %matplotlib inline
      import matplotlib
      import seaborn as sns
      from plotly.subplots import make_subplots
      import plotly.graph_objects as go
```

- **pandas:** For data handling.
- **numpy:** For numerical computations.
- **matplotlib.pyplot and seaborn:** For creating visualizations.

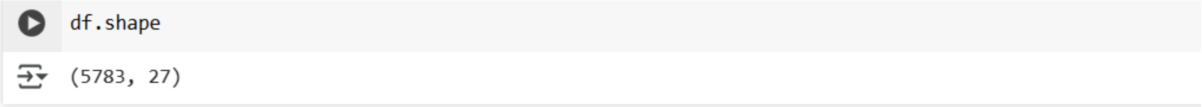
4.2 Reading the Dataset

importing the csv file as a dataframe

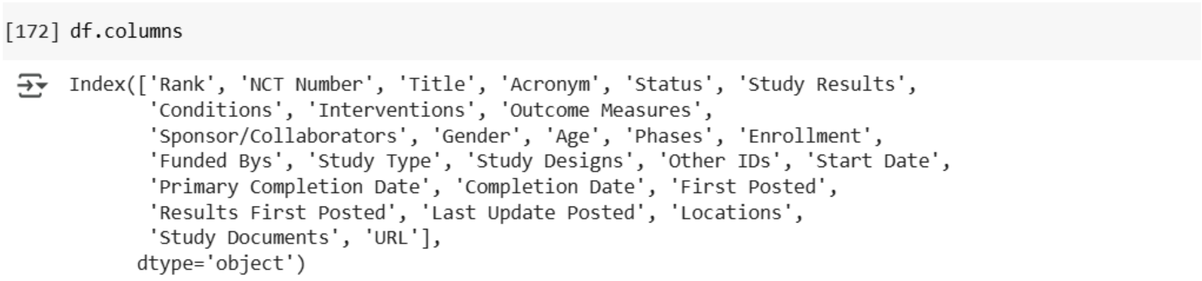


```
[127] df = pd.read_csv('/content/COVID clinical trials.csv')
```

4.3 Checking Basic Information

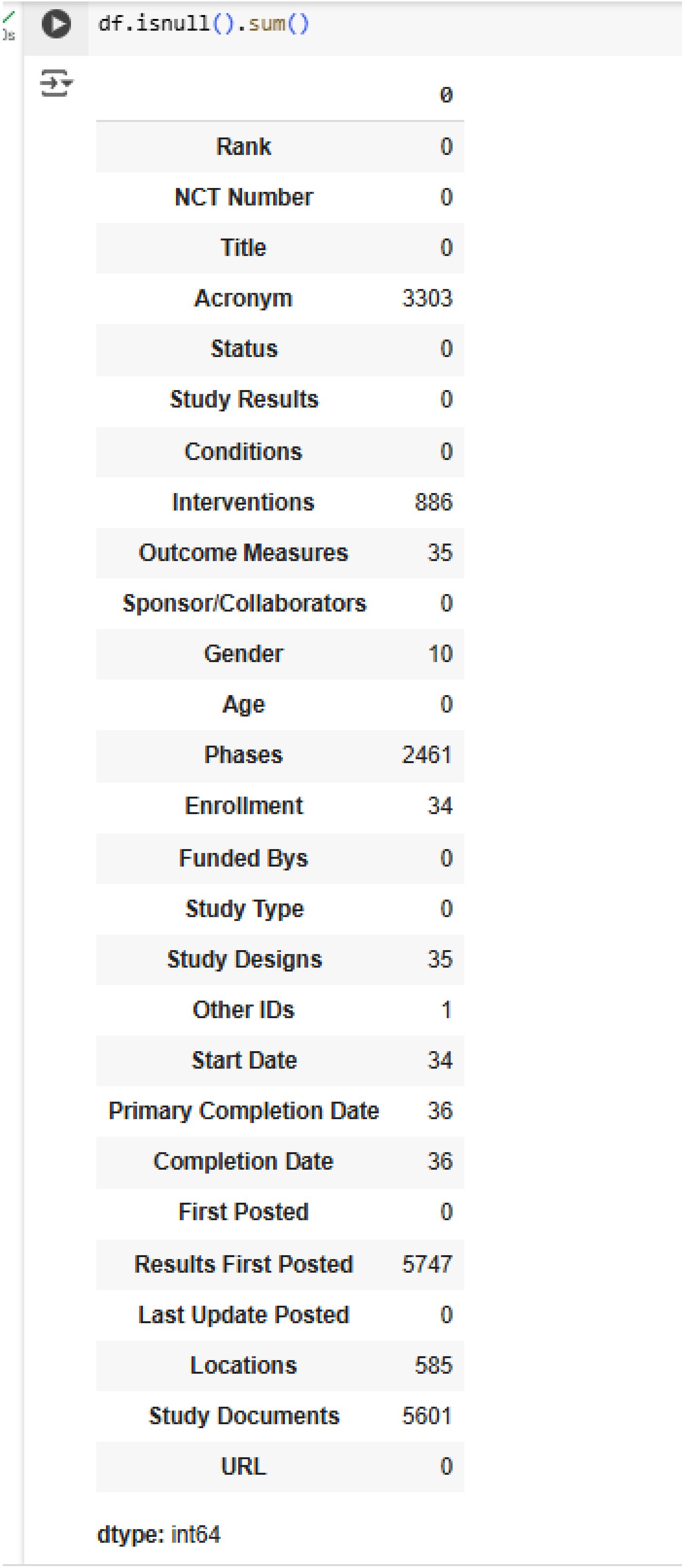


```
df.shape
(5783, 27)
```



```
[172] df.columns

Index(['Rank', 'NCT Number', 'Title', 'Acronym', 'Status', 'Study Results',
       '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',
       'Results First Posted', 'Last Update Posted', 'Locations',
       'Study Documents', 'URL'],
      dtype='object')
```

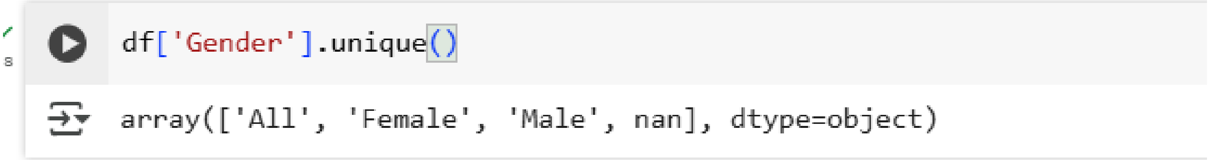
4.4 Data Cleaning



```python
df.isnull().sum()
```

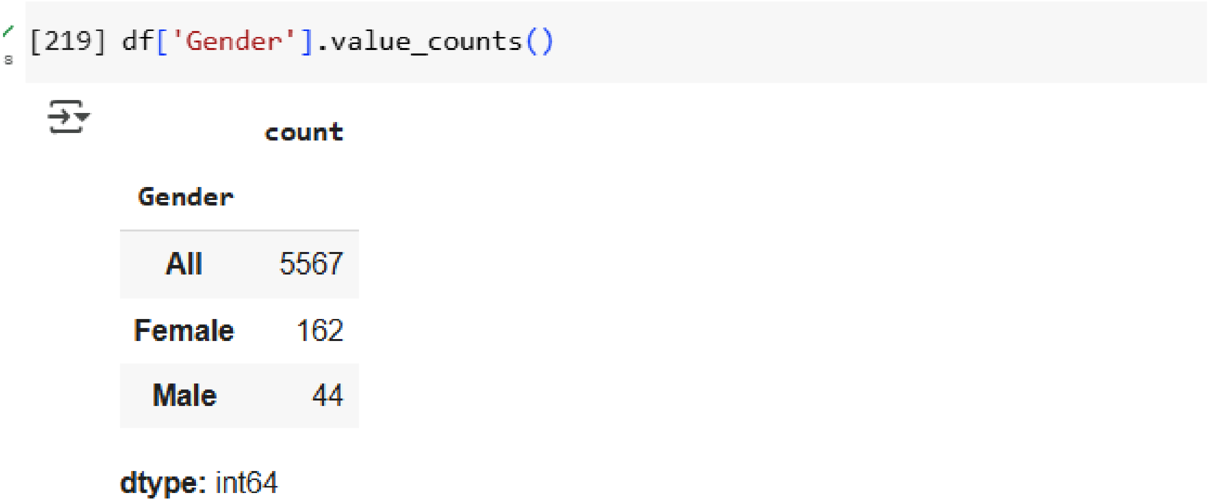
```
                              0
                 Rank         0
           NCT Number         0
                Title         0
              Acronym      3303
               Status         0
        Study Results         0
           Conditions         0
        Interventions       886
     Outcome Measures        35
Sponsor/Collaborators         0
               Gender        10
                  Age         0
               Phases      2461
           Enrollment        34
          Funded Bys          0
           Study Type         0
        Study Designs        35
            Other IDs         1
           Start Date        34
Primary Completion Date      36
      Completion Date        36
         First Posted         0
  Results First Posted     5747
    Last Update Posted        0
            Locations       585
      Study Documents      5601
                  URL         0

dtype: int64
```

4.5 Explore the Gender distribution in the studies



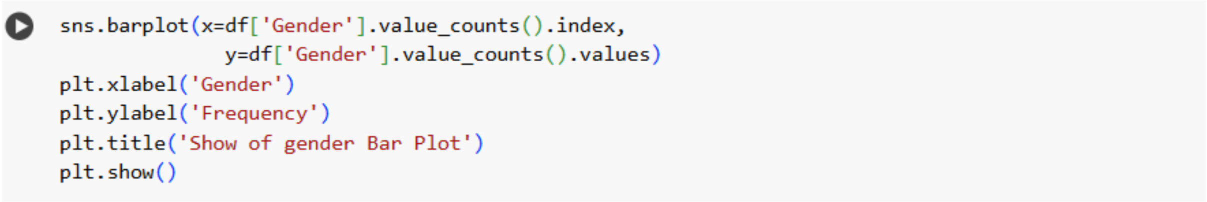
```
df['Gender'].unique()
```

```
array(['All', 'Female', 'Male', nan], dtype=object)
```

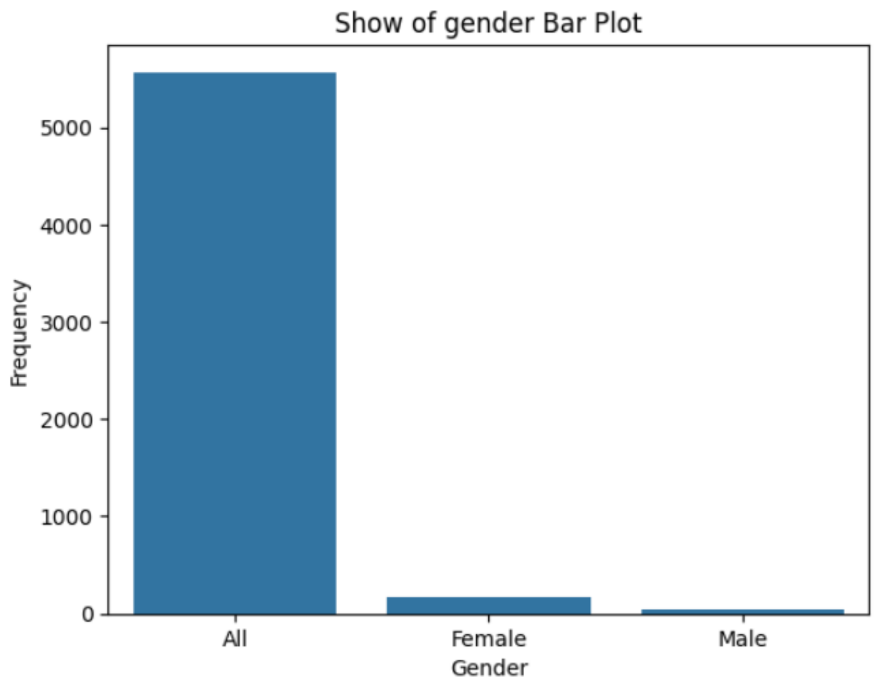


```
[219] df['Gender'].value_counts()
```

```
          count
Gender
All        5567
Female      162
Male         44

dtype: int64
```

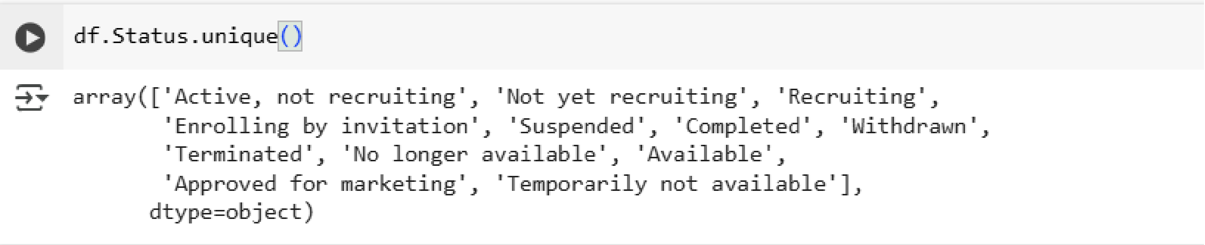


```
sns.barplot(x=df['Gender'].value_counts().index,
            y=df['Gender'].value_counts().values)
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.title('Show of gender Bar Plot')
plt.show()
```



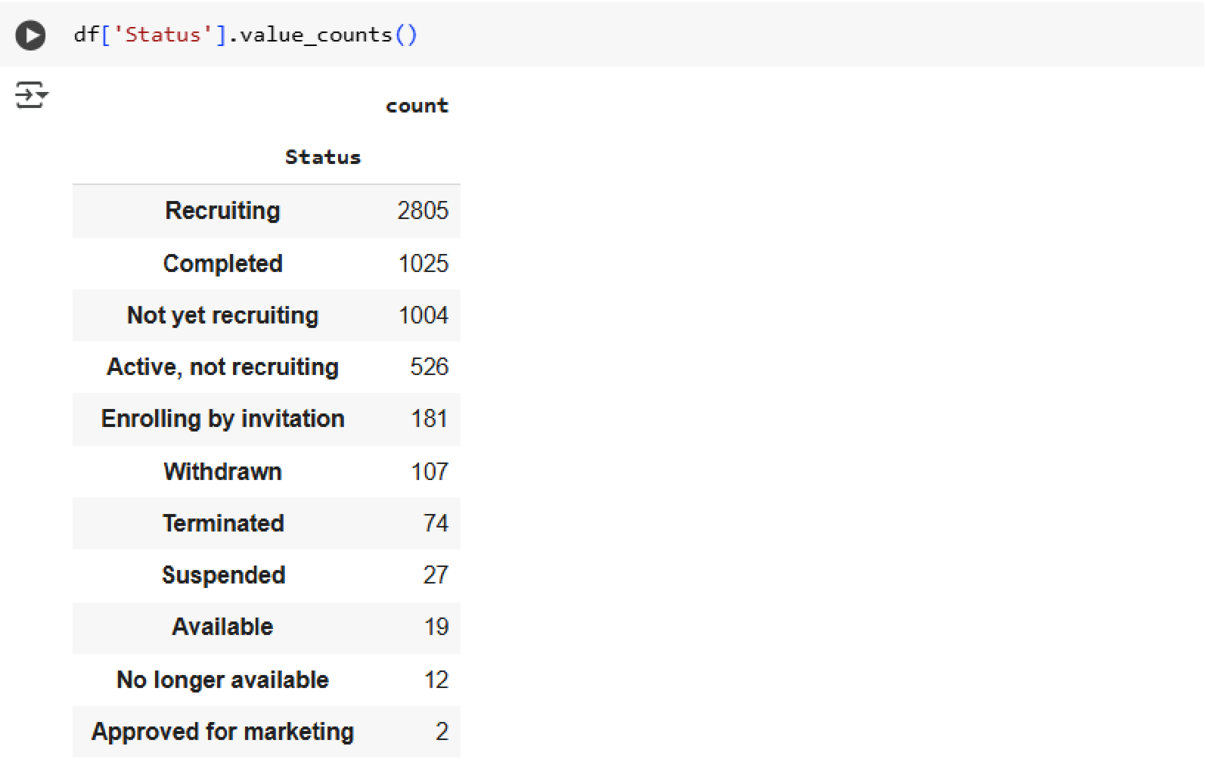
Observation: The majority of studies are categorized as "All" genders, with few specifically labeled as "Female" or "Male."

4.6 Exploring Study Status Distribution



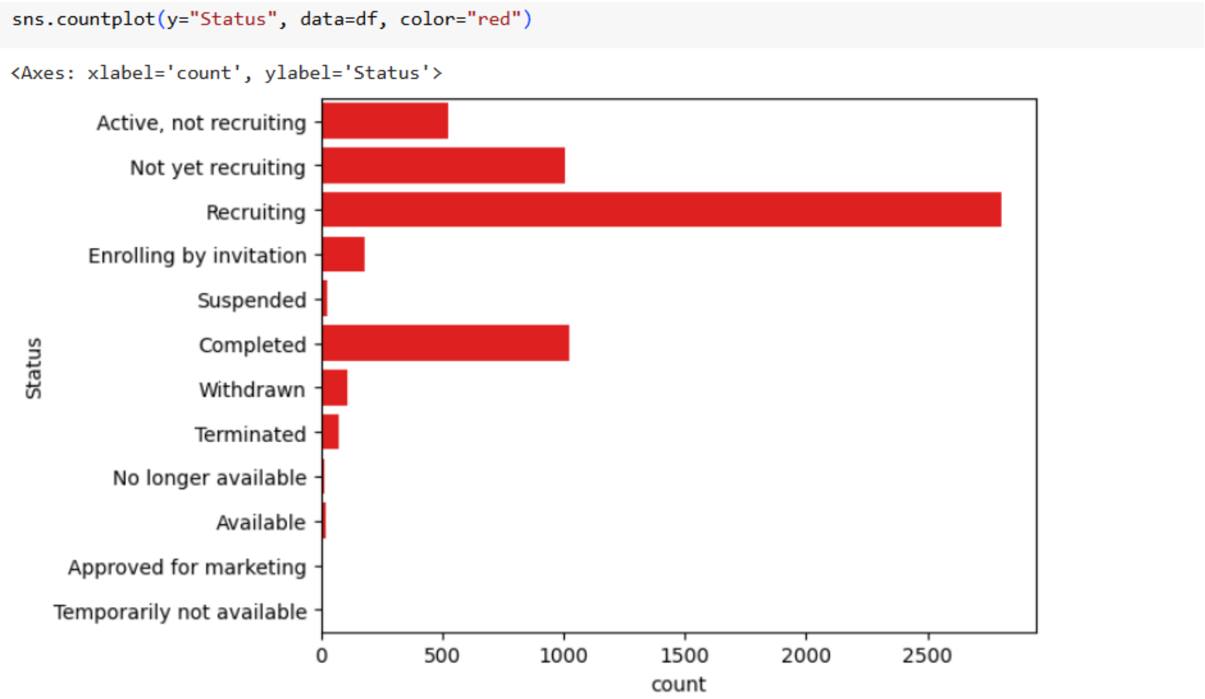
```
df.Status.unique()
```

```
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)
```



```
df['Status'].value_counts()
```

	count
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

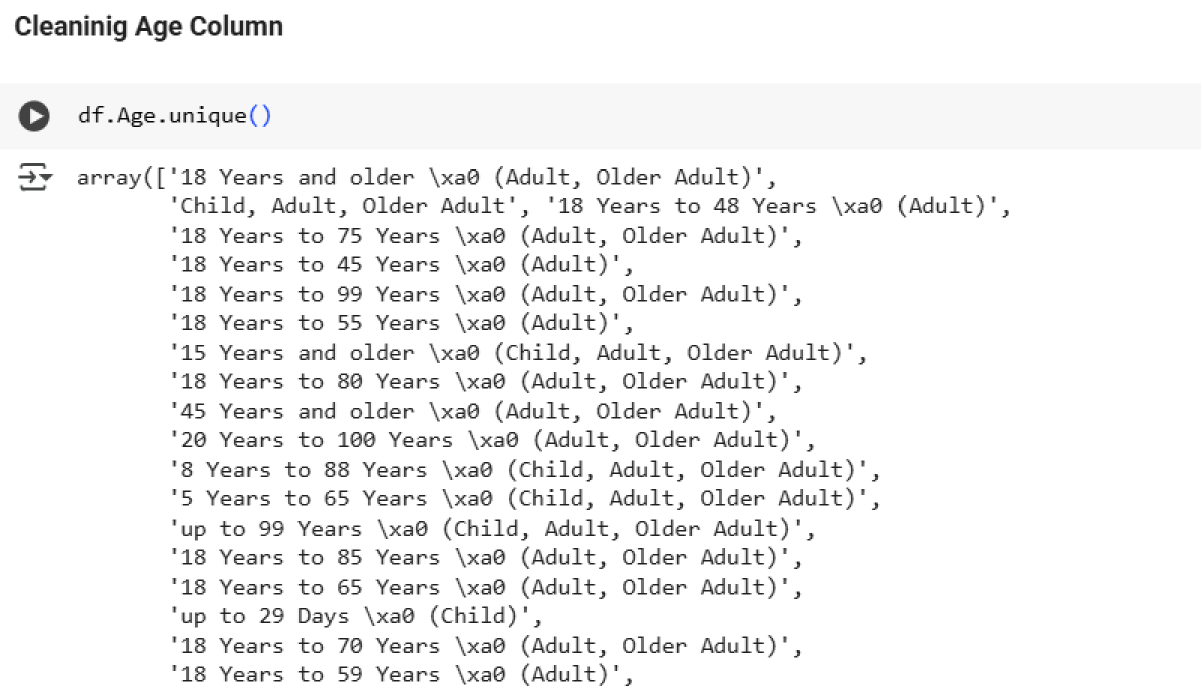


```
sns.countplot(y="Status", data=df, color="red")
```

```
<Axes: xlabel='count', ylabel='Status'>
```

Observation: The study status distribution reveals that most studies are still in the process of recruiting participants, indicating that the research is ongoing.

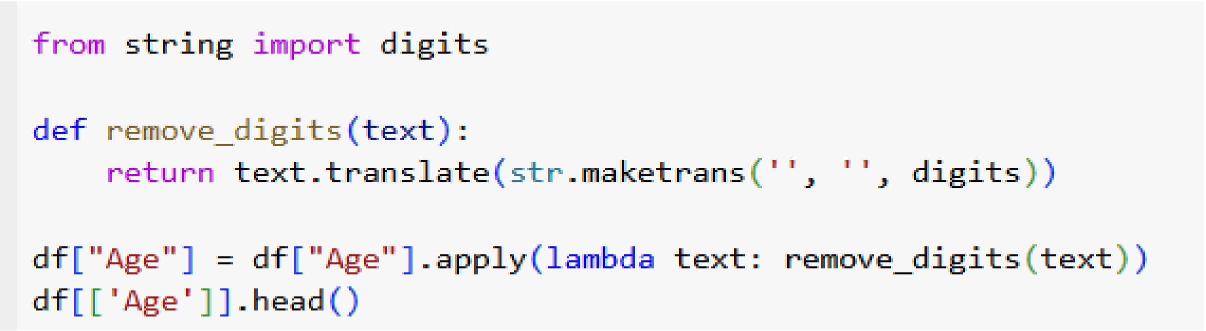
4.7 Cleaning Age Column



Cleaninig Age Column

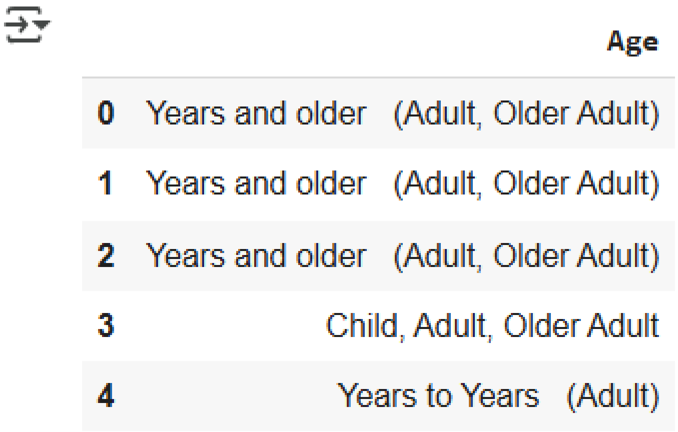
```
df.Age.unique()
```

```
array(['18 Years and older \xa0 (Adult, Older Adult)',
       'Child, Adult, Older Adult', '18 Years to 48 Years \xa0 (Adult)',
       '18 Years to 75 Years \xa0 (Adult, Older Adult)',
       '18 Years to 45 Years \xa0 (Adult)',
       '18 Years to 99 Years \xa0 (Adult, Older Adult)',
       '18 Years to 55 Years \xa0 (Adult)',
       '15 Years and older \xa0 (Child, Adult, Older Adult)',
       '18 Years to 80 Years \xa0 (Adult, Older Adult)',
       '45 Years and older \xa0 (Adult, Older Adult)',
       '20 Years to 100 Years \xa0 (Adult, Older Adult)',
       '8 Years to 88 Years \xa0 (Child, Adult, Older Adult)',
       '5 Years to 65 Years \xa0 (Child, Adult, Older Adult)',
       'up to 99 Years \xa0 (Child, Adult, Older Adult)',
       '18 Years to 85 Years \xa0 (Adult, Older Adult)',
       '18 Years to 65 Years \xa0 (Adult, Older Adult)',
       'up to 29 Days \xa0 (Child)',
       '18 Years to 70 Years \xa0 (Adult, Older Adult)',
       '18 Years to 59 Years \xa0 (Adult)',
```

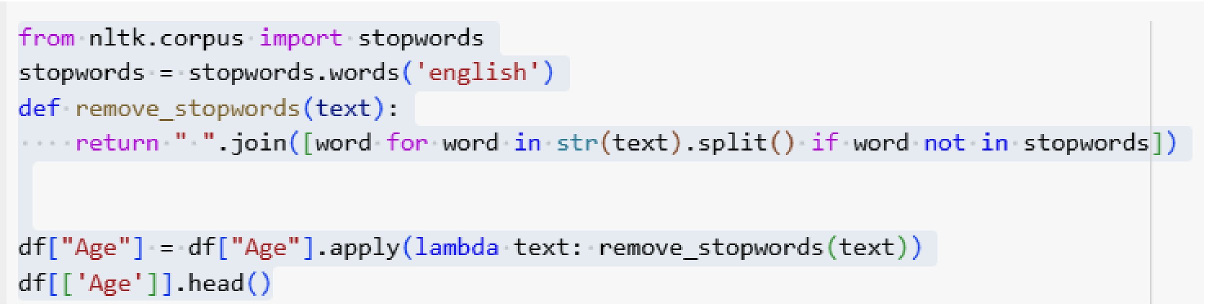


```
from string import digits

def remove_digits(text):
    return text.translate(str.maketrans('', '', digits))

df["Age"] = df["Age"].apply(lambda text: remove_digits(text))
df[['Age']].head()
```

- **from string import digits**: Imports the digits constant, which contains all digits ('0123456789').
- **remove_digits(text)**: A function that removes all digit characters from a given string text by using str.translate() to replace digits (from digits) with nothing (i.e., removes them).
- **df["Age"].apply(lambda text: remove_digits(text))**: For each value in the "Age" column of the df DataFrame, the remove_digits function is applied to remove any digits.
- **df[['Age']].head()**: Displays the first 5 rows of the "Age" column after removing digits.

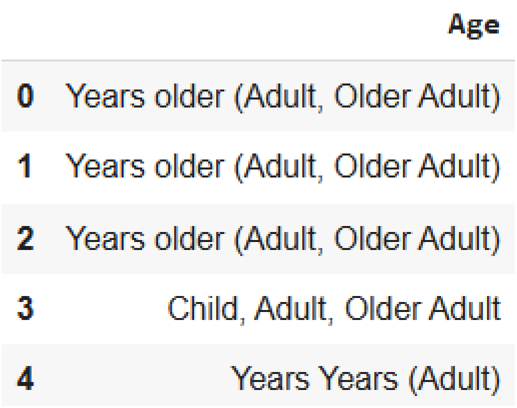


	Age
0	Years and older (Adult, Older Adult)
1	Years and older (Adult, Older Adult)
2	Years and older (Adult, Older Adult)
3	Child, Adult, Older Adult
4	Years to Years (Adult)

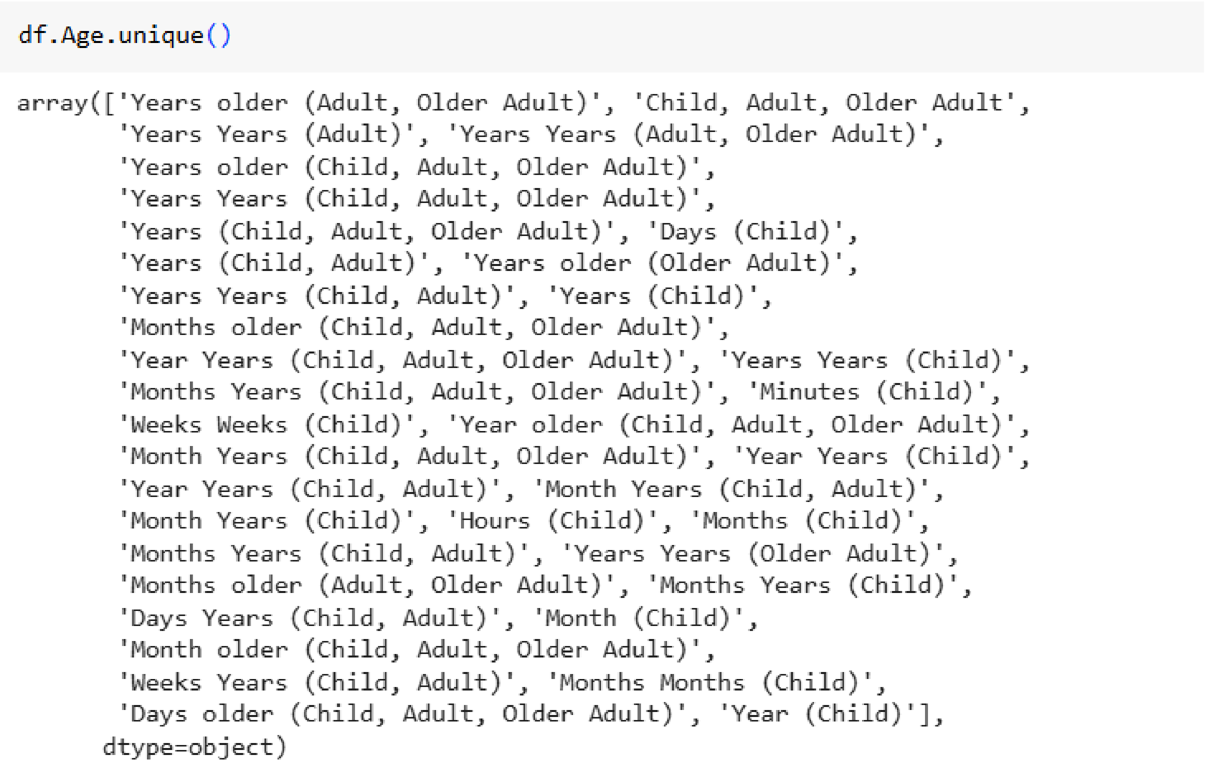


```
from nltk.corpus import stopwords
stopwords = stopwords.words('english')
def remove_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in stopwords])

df["Age"] = df["Age"].apply(lambda text: remove_stopwords(text))
df[['Age']].head()
```

- **from nltk.corpus import stopwords**: Imports common English stopwords (like "the", "is", "and") from NLTK.
- **remove_stopwords(text)**: Defines a function that removes these stopwords from a given text by keeping only the words not in the stopwords list.
- **df["Age"].apply(lambda text: remove_stopwords(text))**: Applies the remove_stopwords function to each entry in the "Age" column.
- **df[['Age']].head()**: Displays the first 5 rows of the cleaned "Age" column.

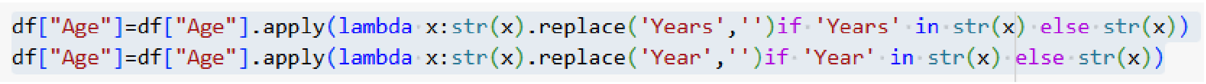


	Age
0	Years older (Adult, Older Adult)
1	Years older (Adult, Older Adult)
2	Years older (Adult, Older Adult)
3	Child, Adult, Older Adult
4	Years Years (Adult)



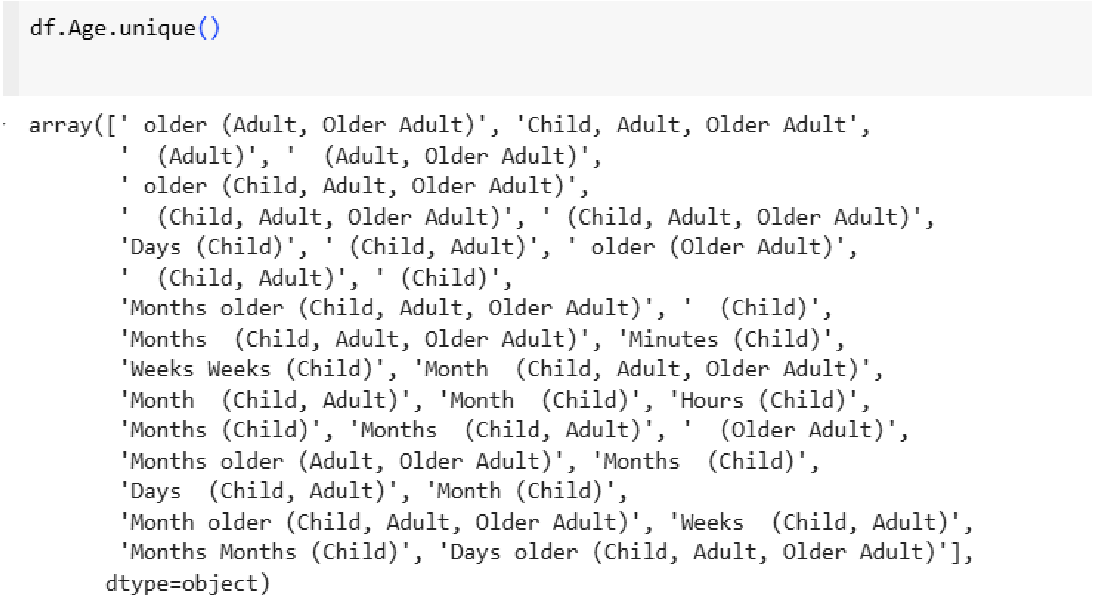
```
df.Age.unique()
```

```
array(['Years older (Adult, Older Adult)', 'Child, Adult, Older Adult',
       'Years Years (Adult)', 'Years Years (Adult, Older Adult)',
       'Years older (Child, Adult, Older Adult)',
       'Years Years (Child, Adult, Older Adult)',
       'Years (Child, Adult, Older Adult)', 'Days (Child)',
       'Years (Child, Adult)', 'Years older (Older Adult)',
       'Years Years (Child, Adult)', 'Years (Child)',
       'Months older (Child, Adult, Older Adult)',
       'Year Years (Child, Adult, Older Adult)', 'Years Years (Child)',
       'Months Years (Child, Adult, Older Adult)', 'Minutes (Child)',
       'Weeks Weeks (Child)', 'Year older (Child, Adult, Older Adult)',
       'Month Years (Child, Adult, Older Adult)', 'Year Years (Child)',
       'Year Years (Child, Adult)', 'Month Years (Child, Adult)',
       'Month Years (Child)', 'Hours (Child)', 'Months (Child)',
       'Months Years (Child, Adult)', 'Years Years (Older Adult)',
       'Months older (Adult, Older Adult)', 'Months Years (Child)',
       'Days Years (Child, Adult)', 'Month (Child)',
       'Month older (Child, Adult, Older Adult)',
       'Weeks Years (Child, Adult)', 'Months Months (Child)',
       'Days older (Child, Adult, Older Adult)', 'Year (Child)'],
      dtype=object)
```



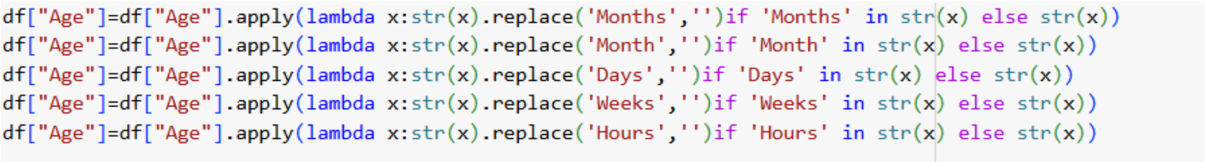
```
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Years','')if 'Years' in str(x) else str(x))
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Year','')if 'Year' in str(x) else str(x))
```

- This code removes the words **"Years"** and **"Year"** from each value in the **"Age"** column.
- It checks if "Years" or "Year" exists in the text, and if yes, replaces them with an empty string.



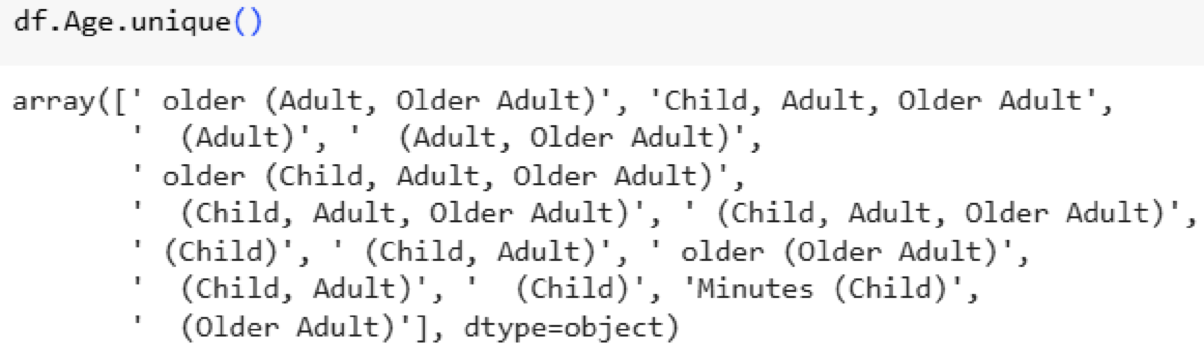
```
df.Age.unique()
```

```
array([' older (Adult, Older Adult)', 'Child, Adult, Older Adult',
       '  (Adult)', '  (Adult, Older Adult)',
       ' older (Child, Adult, Older Adult)',
       '  (Child, Adult, Older Adult)', ' (Child, Adult, Older Adult)',
       'Days (Child)', ' (Child, Adult)', ' older (Older Adult)',
       '  (Child, Adult)', ' (Child)',
       'Months older (Child, Adult, Older Adult)', '  (Child)',
       'Months  (Child, Adult, Older Adult)', 'Minutes (Child)',
       'Weeks Weeks (Child)', 'Month  (Child, Adult, Older Adult)',
       'Month  (Child, Adult)', 'Month  (Child)', 'Hours (Child)',
       'Months (Child)', 'Months  (Child, Adult)', '  (Older Adult)',
       'Months older (Adult, Older Adult)', 'Months  (Child)',
       'Days  (Child, Adult)', 'Month (Child)',
       'Month older (Child, Adult, Older Adult)', 'Weeks  (Child, Adult)',
       'Months Months (Child)', 'Days older (Child, Adult, Older Adult)'],
      dtype=object)
```



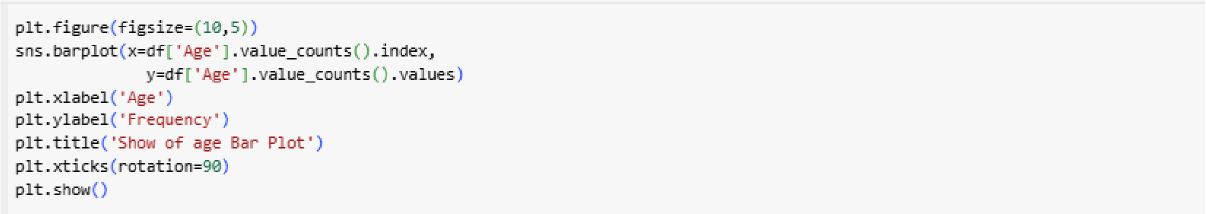
```
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Months','')if 'Months' in str(x) else str(x))
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Month','')if 'Month' in str(x) else str(x))
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Days','')if 'Days' in str(x) else str(x))
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Weeks','')if 'Weeks' in str(x) else str(x))
df["Age"]=df["Age"].apply(lambda x:str(x).replace('Hours','')if 'Hours' in str(x) else str(x))
```

- This code **removes** the words **"Months"**, **"Month"**, **"Days"**, **"Weeks"**, and **"Hours"** from each value in the **"Age"** column.
- For each word, it checks if it exists in the text and replaces it with an empty string.

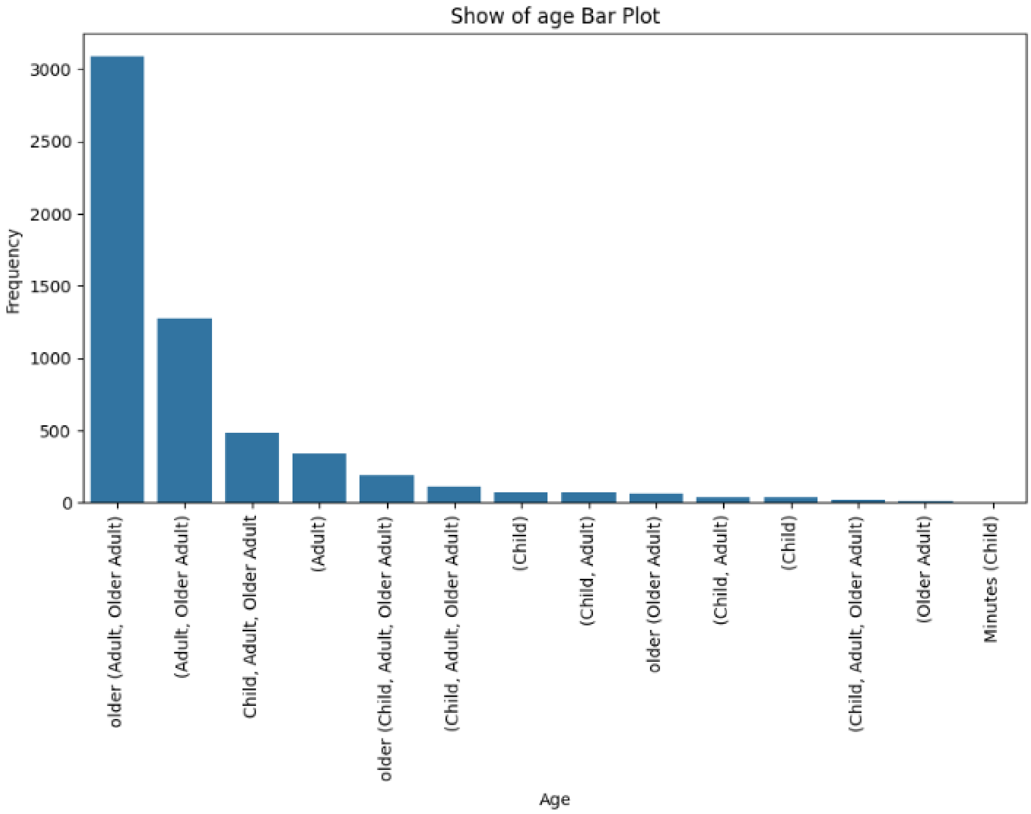


```
df.Age.unique()
```

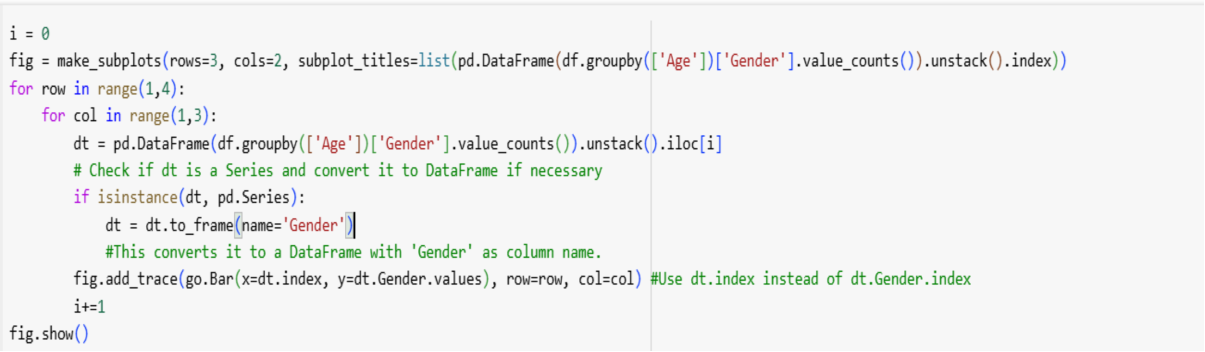
```
array([' older (Adult, Older Adult)', 'Child, Adult, Older Adult',
       '  (Adult)', '  (Adult, Older Adult)',
       ' older (Child, Adult, Older Adult)',
       '  (Child, Adult, Older Adult)', ' (Child, Adult, Older Adult)',
       ' (Child)', ' (Child, Adult)', ' older (Older Adult)',
       '  (Child, Adult)', '  (Child)', 'Minutes (Child)',
       '  (Older Adult)'], dtype=object)
```



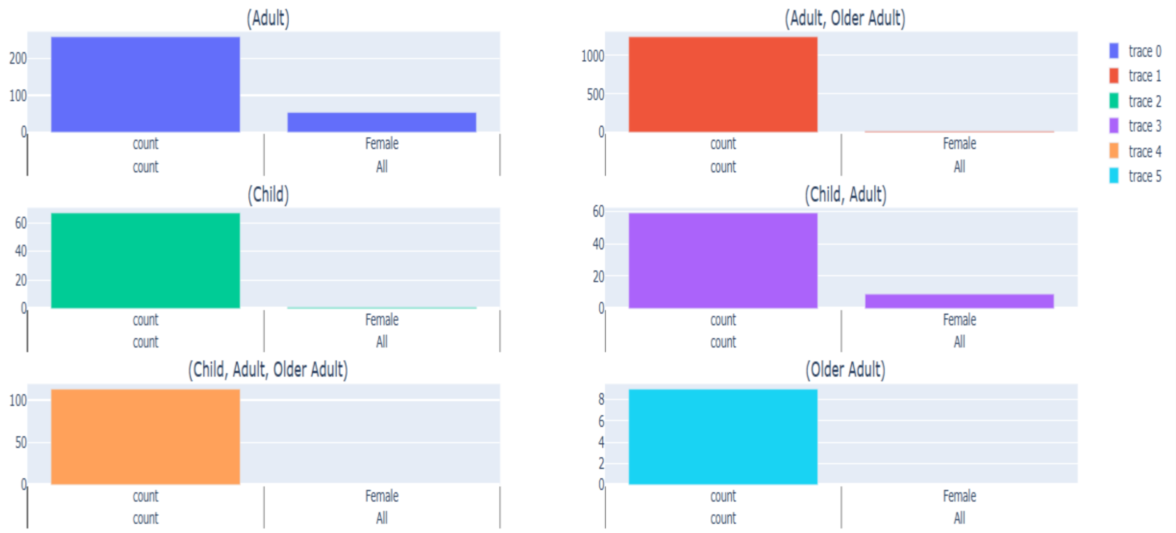
```
plt.figure(figsize=(10,5))
sns.barplot(x=df['Age'].value_counts().index,
              y=df['Age'].value_counts().values)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Show of age Bar Plot')
plt.xticks(rotation=90)
plt.show()
```



Observation: Adult, Older Adult age bracket are mostly studied --Child and older Adult age bracket has the lowest studies.



```
i = 0
fig = make_subplots(rows=3, cols=2, subplot_titles=list(pd.DataFrame(df.groupby(['Age'])['Gender'].value_counts()).unstack().index))
for row in range(1,4):
    for col in range(1,3):
        dt = pd.DataFrame(df.groupby(['Age'])['Gender'].value_counts()).unstack().iloc[i]
        # Check if dt is a Series and convert it to DataFrame if necessary
        if isinstance(dt, pd.Series):
            dt = dt.to_frame(name='Gender')
            #This converts it to a DataFrame with 'Gender' as column name.
        fig.add_trace(go.Bar(x=dt.index, y=dt.Gender.values), row=row, col=col) #Use dt.index instead of dt.Gender.index
        i+=1
fig.show()
```

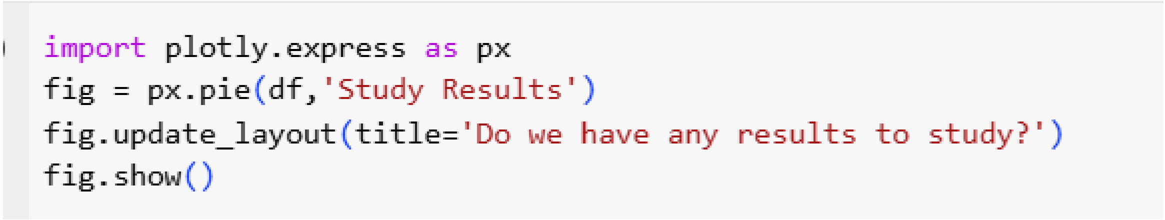


Observations:

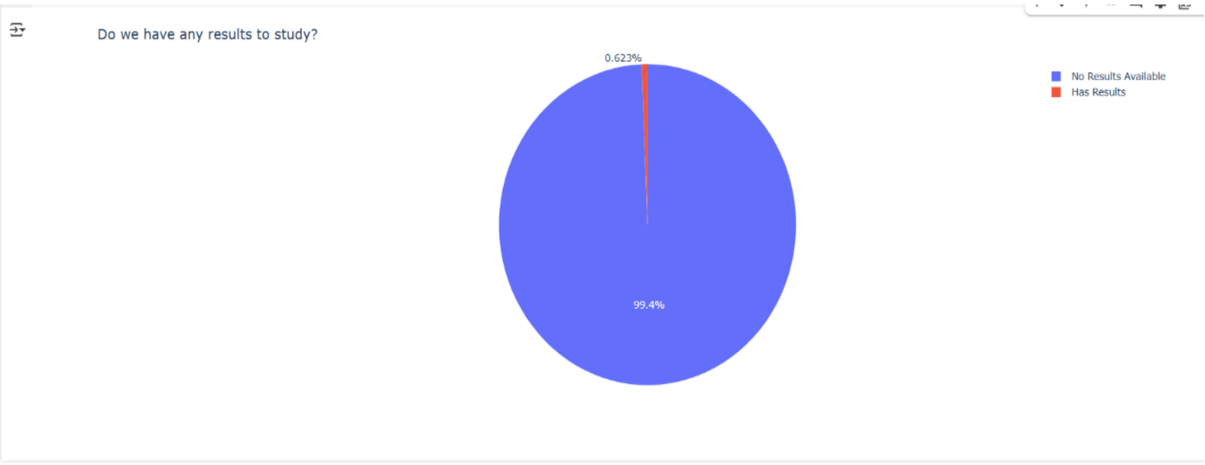
Most studies have taken data from All Genders;

In (Adult) and (Child, Adult) Category there is significant number of Female patients considered for the studies

4.8 Exploring study results



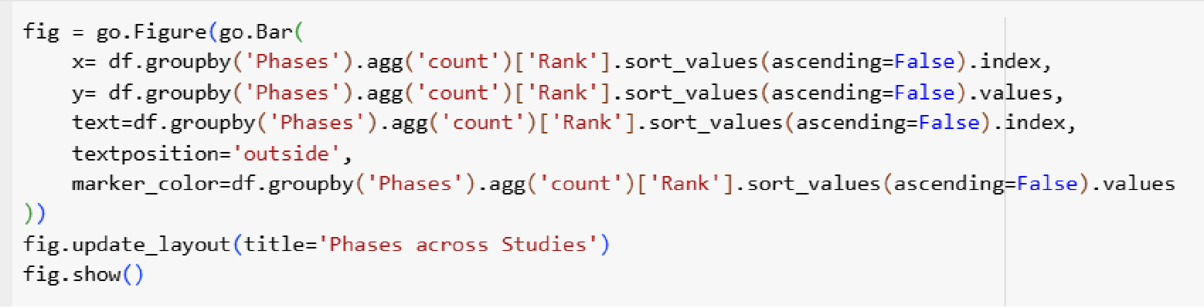
```
import plotly.express as px
fig = px.pie(df,'Study Results')
fig.update_layout(title='Do we have any results to study?')
fig.show()
```



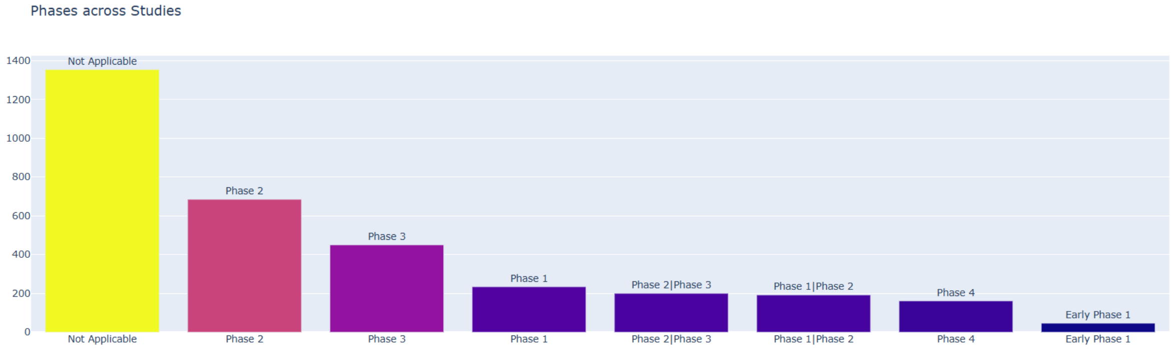
0.038% studies have attained some results;

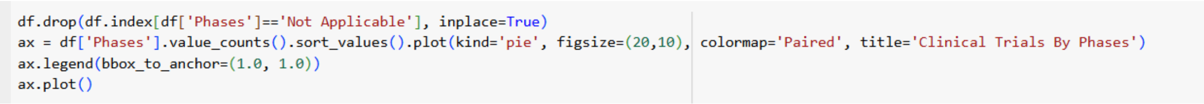
Remaining 99.97% results have NO RESULTS

4.9 Exploring Study Phases

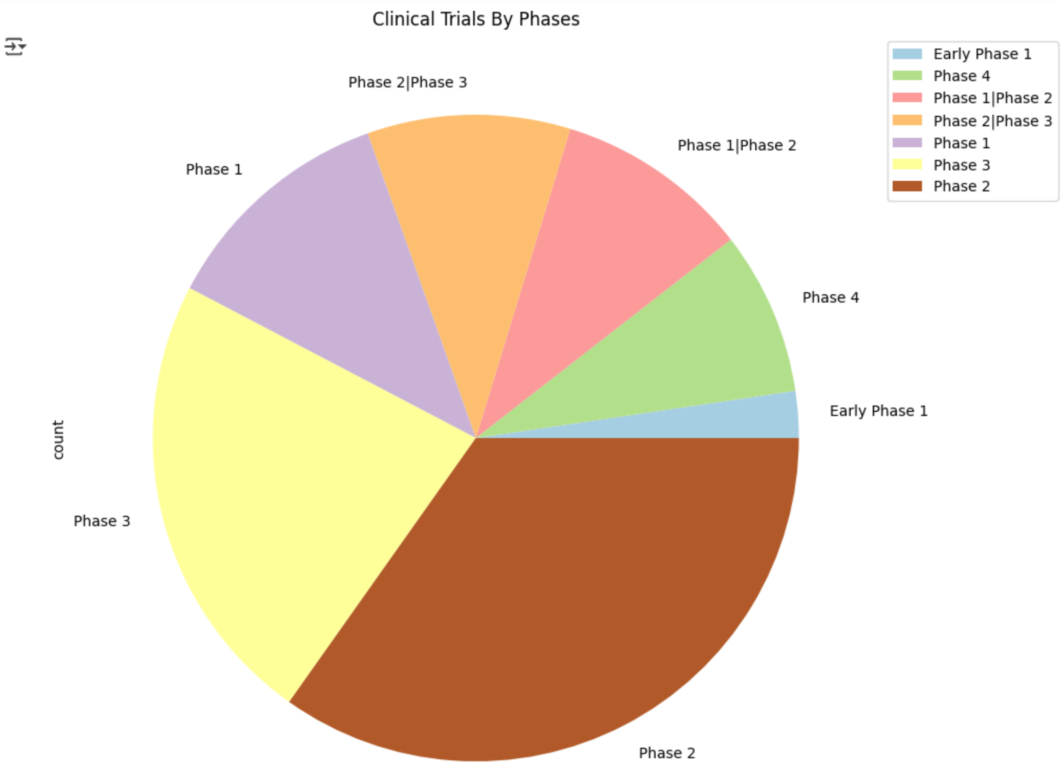


```
fig = go.Figure(go.Bar(
    x= df.groupby('Phases').agg('count')['Rank'].sort_values(ascending=False).index,
    y= df.groupby('Phases').agg('count')['Rank'].sort_values(ascending=False).values,
    text=df.groupby('Phases').agg('count')['Rank'].sort_values(ascending=False).index,
    textposition='outside',
    marker_color=df.groupby('Phases').agg('count')['Rank'].sort_values(ascending=False).values
))
fig.update_layout(title='Phases across Studies')
fig.show()
```

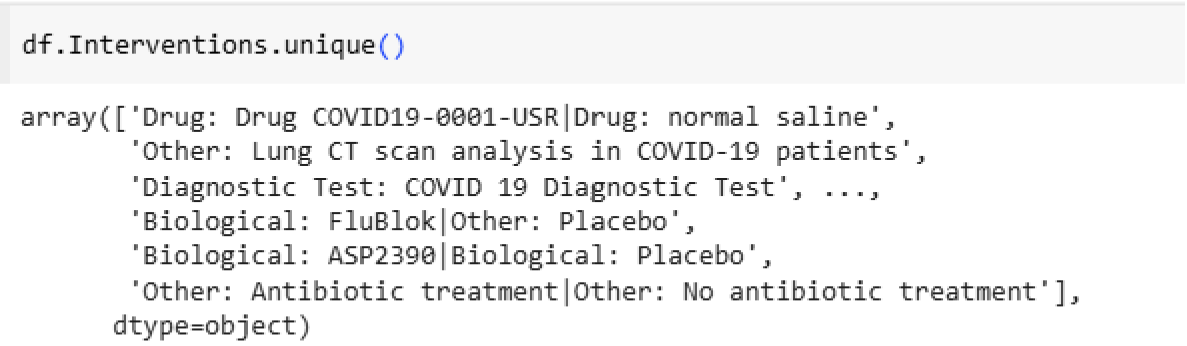




```
df.drop(df.index[df['Phases']=='Not Applicable'], inplace=True)
ax = df['Phases'].value_counts().sort_values().plot(kind='pie', figsize=(20,10), colormap='Paired', title='Clinical Trials By Phases')
ax.legend(bbox_to_anchor=(1.0, 1.0))
ax.plot()
```

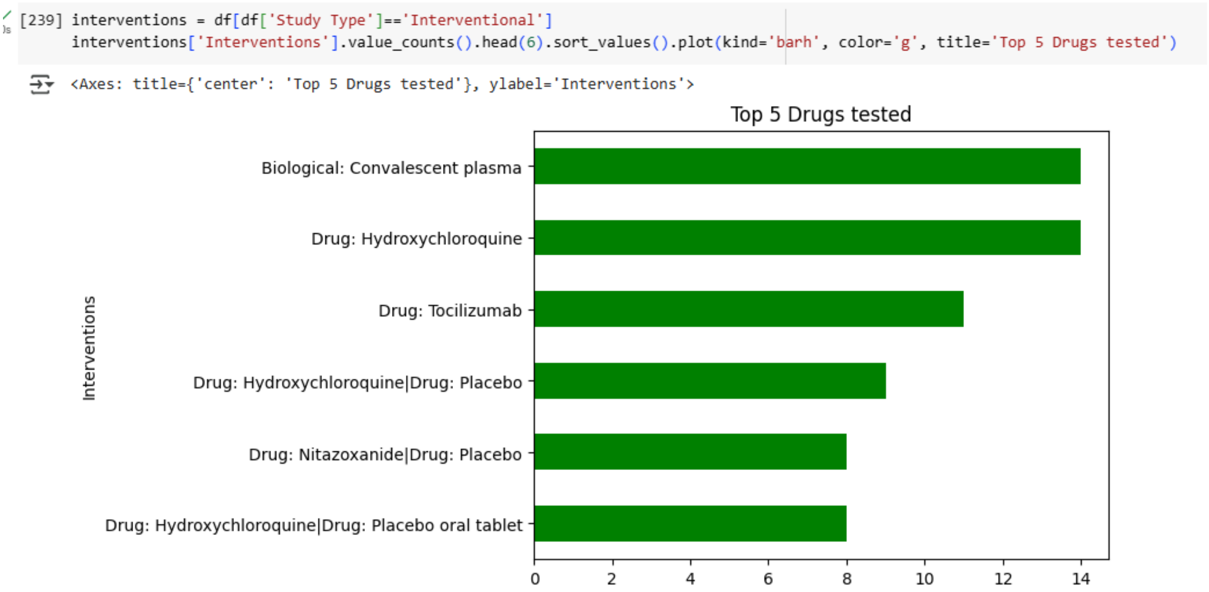


Observation: Most Study where applicable are in Phase 2 and Phase 3



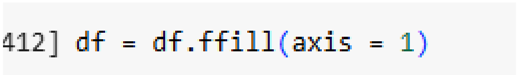
```
df.Interventions.unique()
```

```
array(['Drug: Drug COVID19-0001-USR|Drug: normal saline',
       'Other: Lung CT scan analysis in COVID-19 patients',
       'Diagnostic Test: COVID 19 Diagnostic Test', ...,
       'Biological: FluBlok|Other: Placebo',
       'Biological: ASP2390|Biological: Placebo',
       'Other: Antibiotic treatment|Other: No antibiotic treatment'],
      dtype=object)
```



```
[239] interventions = df[df['Study Type']=='Interventional']
      interventions['Interventions'].value_counts().head(6).sort_values().plot(kind='barh', color='g', title='Top 5 Drugs tested')
```

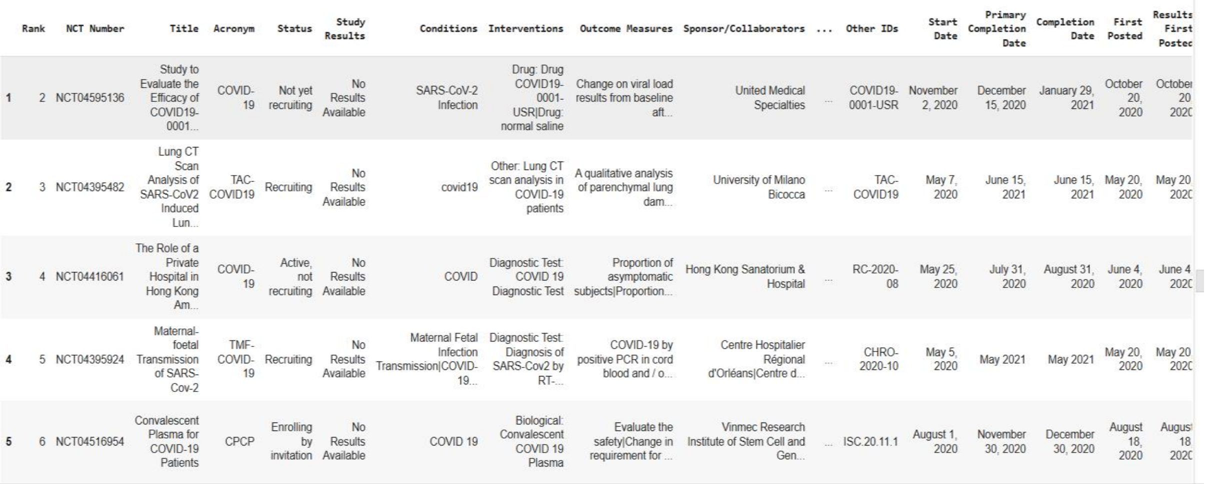
```
<Axes: title={'center': 'Top 5 Drugs tested'}, ylabel='Interventions'>
```



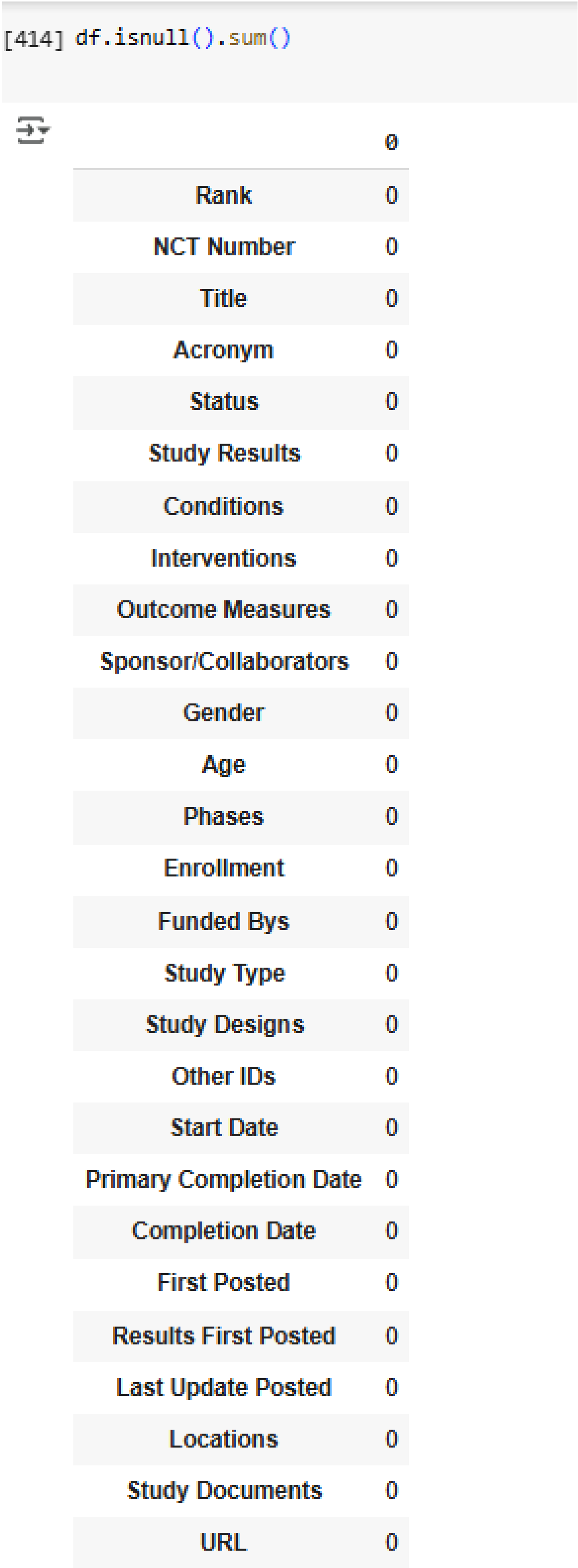
```
412] df = df.ffill(axis = 1)
```

"For each row, if a cell is missing, fill it with the value from the left."

```
df.head()
```

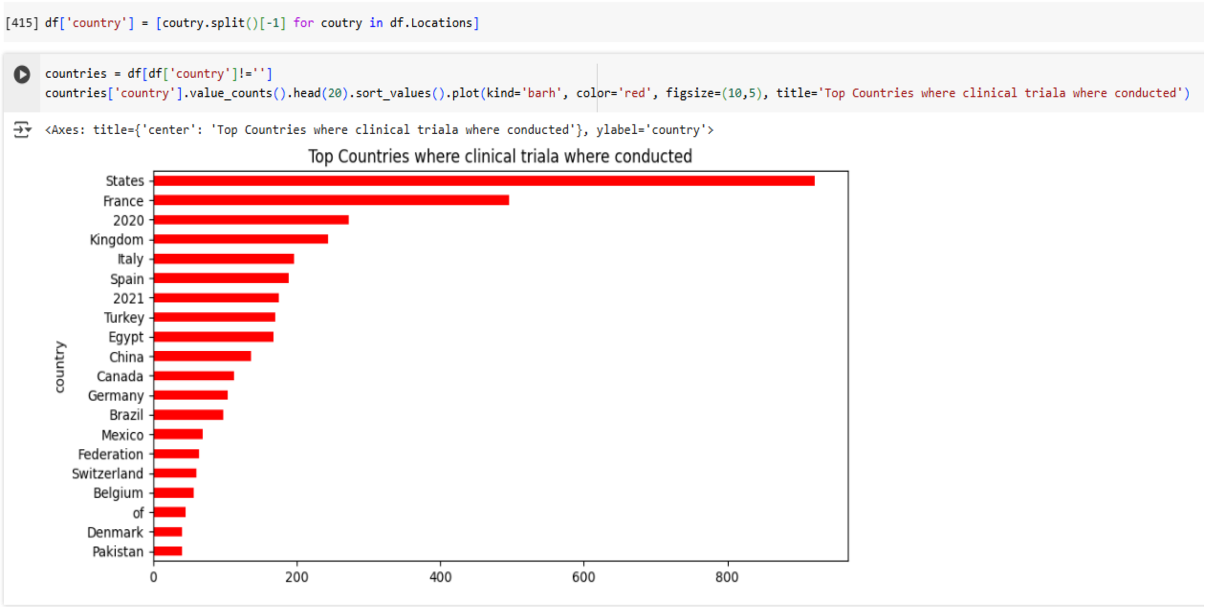


	Rank	NCT Number	Title	Acronym	Status	Study Results	Conditions	Interventions	Outcome Measures	Sponsor/Collaborators	...	Other IDs	Start Date	Primary Completion Date	Completion Date	First Posted	Results First Postec
1	2	NCT04595136	Study to Evaluate the Efficacy of COVID19-0001...	COVID-19	Not yet recruiting	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	Octobe 20, 202
2	3	NCT04395482	Lung CT Scan Analysis of SARS-CoV2 Induced Lun...	TAC-COVID19	Recruiting	No Results Available	covid19	Other: Lung CT scan analysis in COVID-19 patients	A qualitative analysis of parenchymal lung dam...	University of Milano Bicocca	...	TAC-COVID19	May 7, 2020	June 15, 2021	June 15, 2021	May 20, 2020	May 20 202
3	4	NCT04416061	The Role of a Private Hospital in Hong Kong Am...	COVID-19	Active, not recruiting	No Results Available	COVID	Diagnostic Test: COVID 19 Diagnostic Test	Proportion of asymptomatic subjects|Proportion...	Hong Kong Sanatorium & Hospital	...	RC-2020-08	May 25, 2020	July 31, 2020	August 31, 2020	June 4, 2020	June 4 202
4	5	NCT04395924	Maternal-foetal Transmission of SARS-Cov-2	TMF-COVID-19	Recruiting	No Results Available	Maternal Fetal Infection Transmission|COVID-19...	Diagnostic Test: Diagnosis of SARS-Cov2 by RT-...	COVID-19 by positive PCR in cord blood and / o...	Centre Hospitalier Régional d'Orléans|Centre d...	...	CHRO-2020-10	May 5, 2020	May 2021	May 2021	May 20, 2020	May 20 202
5	6	NCT04516954	Convalescent Plasma for COVID-19 Patients	CPCP	Enrolling by invitation	No Results Available	COVID 19	Biological: Convalescent COVID 19 Plasma	Evaluate the safety|Change in requirement for ...	Vinmec Research Institute of Stem Cell and Gen...	...	ISC.20.11.1	August 1, 2020	November 30, 2020	December 30, 2020	August 18, 2020	Augus 18 202



```
[414] df.isnull().sum()
```

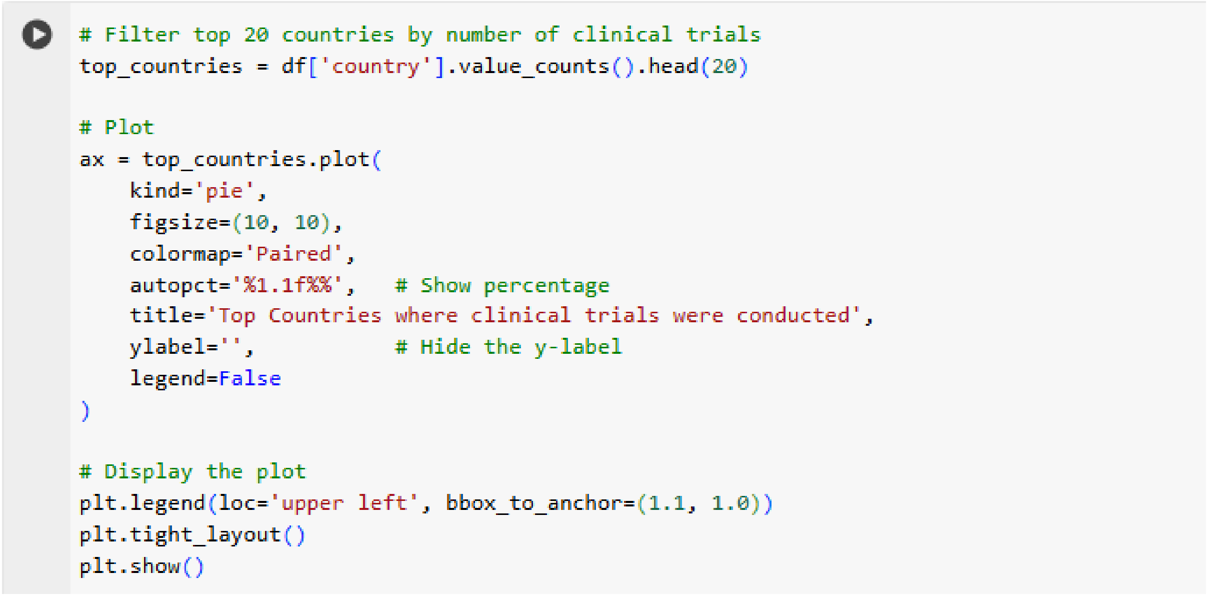
```
                               0
                  Rank         0
            NCT Number         0
                 Title         0
               Acronym         0
                Status         0
         Study Results         0
            Conditions         0
         Interventions         0
      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
  Results First Posted         0
    Last Update Posted         0
             Locations         0
       Study Documents         0
                   URL         0
```



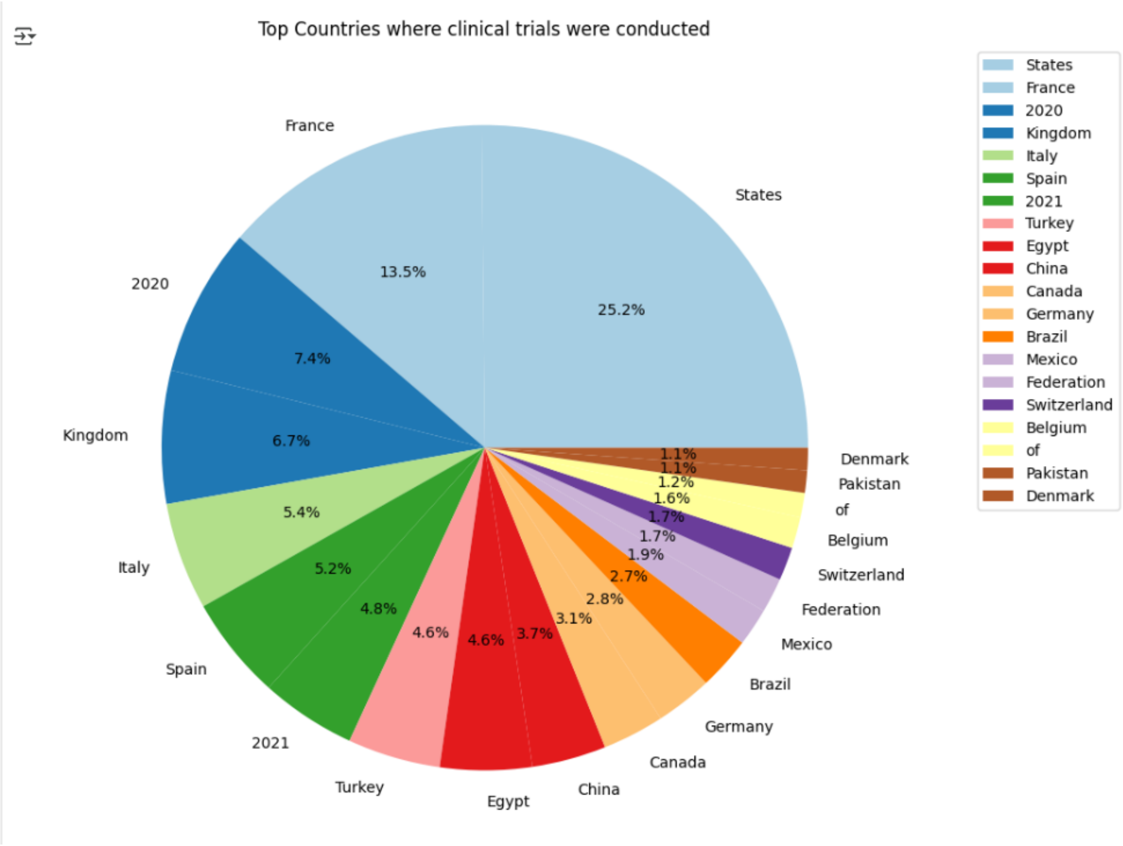
```
[415] df['country'] = [coutry.split()[-1] for coutry in df.Locations]
```

```
countries = df[df['country']!='']
countries['country'].value_counts().head(20).sort_values().plot(kind='barh', color='red', figsize=(10,5), title='Top Countries where clinical triala where conducted')
```

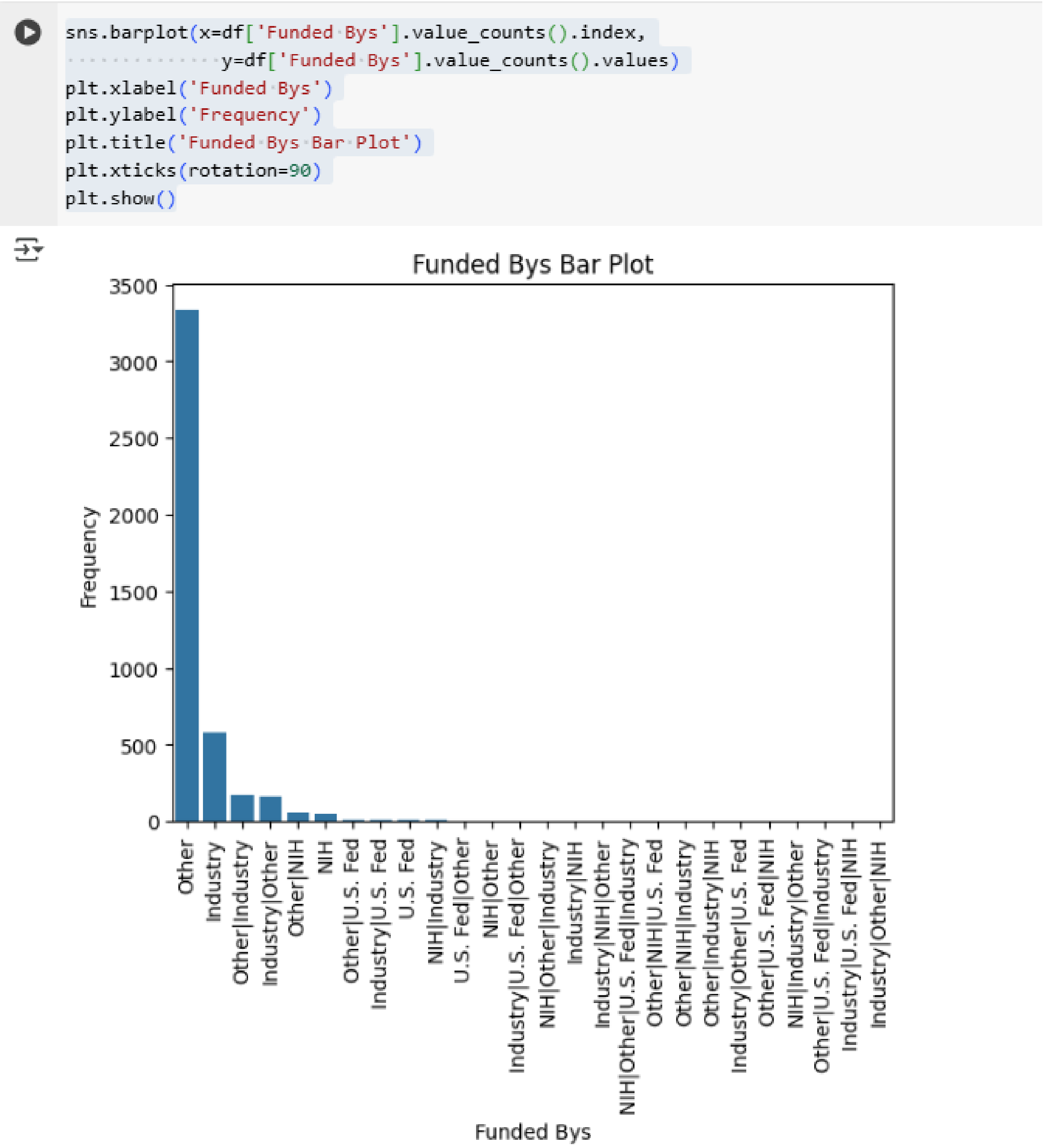
```
<Axes: title={'center': 'Top Countries where clinical triala where conducted'}, ylabel='country'>
```



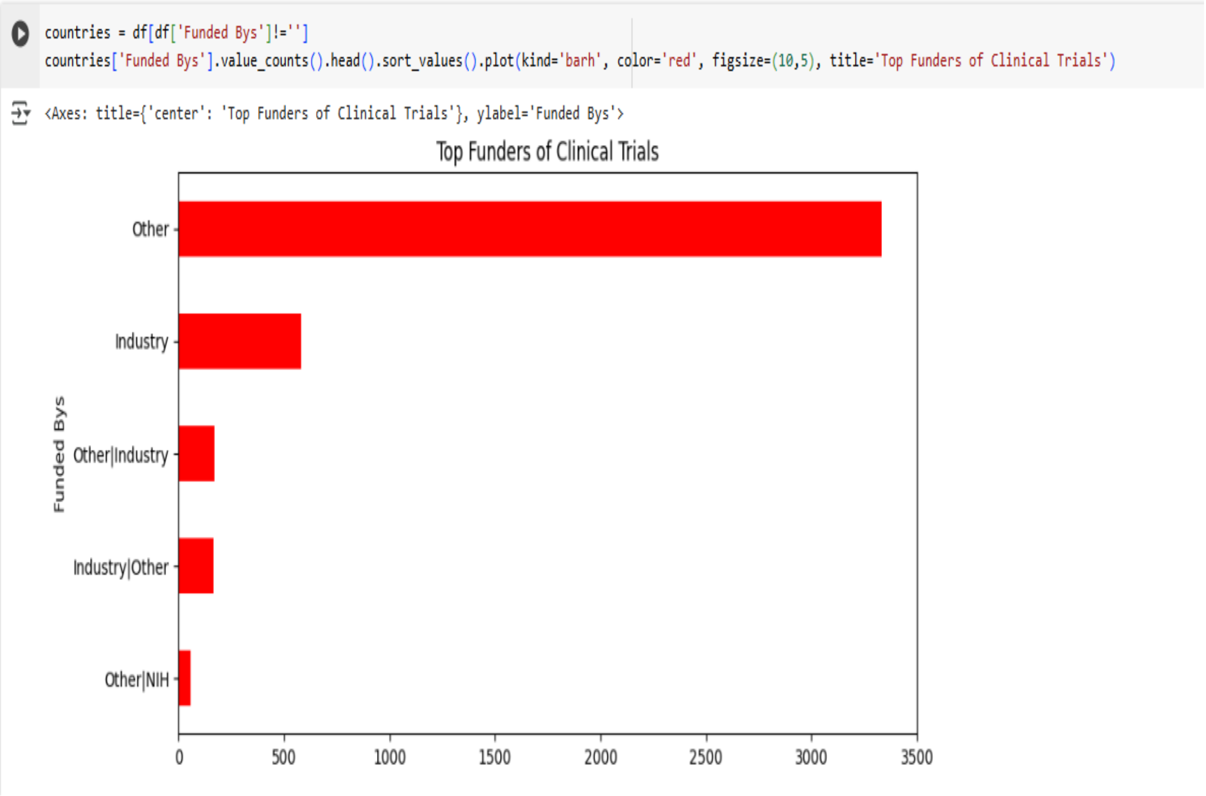
```
# Filter top 20 countries by number of clinical trials
top_countries = df['country'].value_counts().head(20)

# Plot
ax = top_countries.plot(
    kind='pie',
    figsize=(10, 10),
    colormap='Paired',
    autopct='%1.1f%%',   # Show percentage
    title='Top Countries where clinical trials were conducted',
    ylabel='',           # Hide the y-label
    legend=False
)

# Display the plot
plt.legend(loc='upper left', bbox_to_anchor=(1.1, 1.0))
plt.tight_layout()
plt.show()
```



Observation: Most studies have taken data from State and France



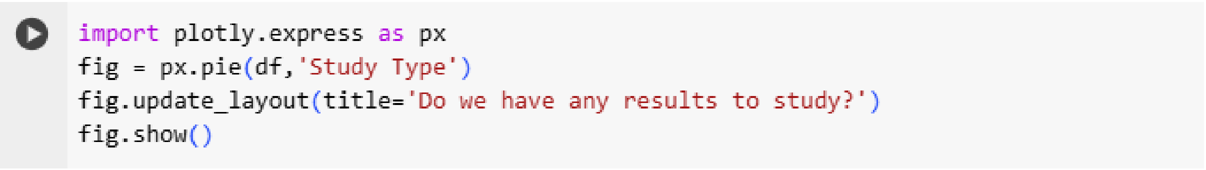
```
sns.barplot(x=df['Funded Bys'].value_counts().index,
              y=df['Funded Bys'].value_counts().values)
plt.xlabel('Funded Bys')
plt.ylabel('Frequency')
plt.title('Funded Bys Bar Plot')
plt.xticks(rotation=90)
plt.show()
```



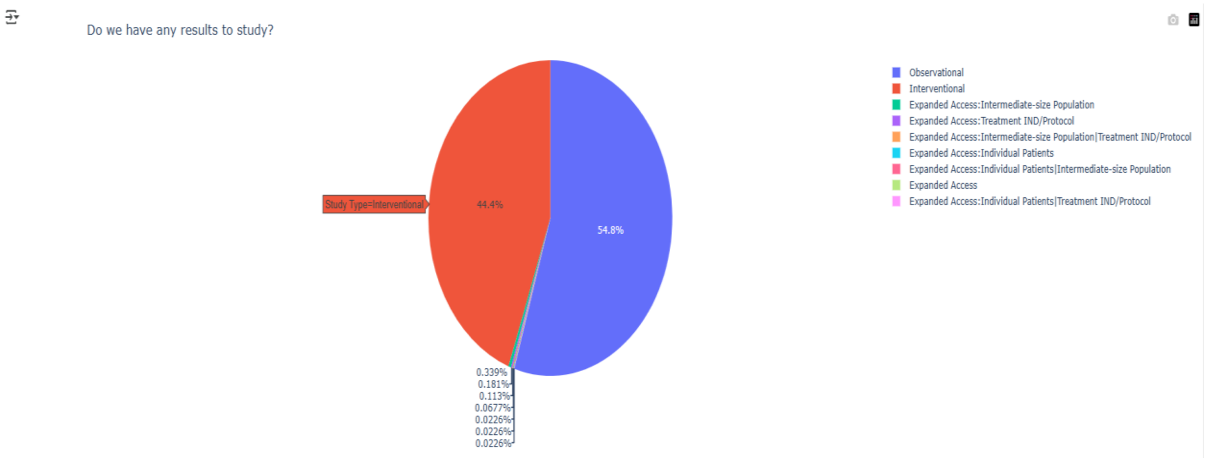
```
countries = df[df['Funded Bys']!='']
countries['Funded Bys'].value_counts().head().sort_values().plot(kind='barh', color='red', figsize=(10,5), title='Top Funders of Clinical Trials')
```

```
<Axes: title={'center': 'Top Funders of Clinical Trials'}, ylabel='Funded Bys'>
```

Observation: Max Funding is by Industry

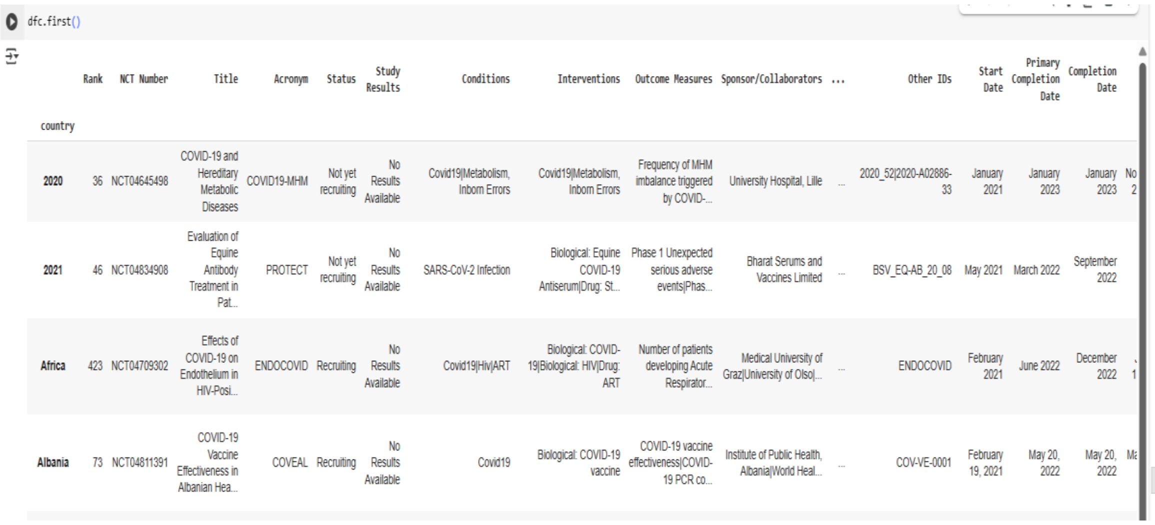


```
import plotly.express as px
fig = px.pie(df,'Study Type')
fig.update_layout(title='Do we have any results to study?')
fig.show()
```



4.10 States Explore the Gender distribution in the studies

```
[637] dfc = df.groupby('country')
```



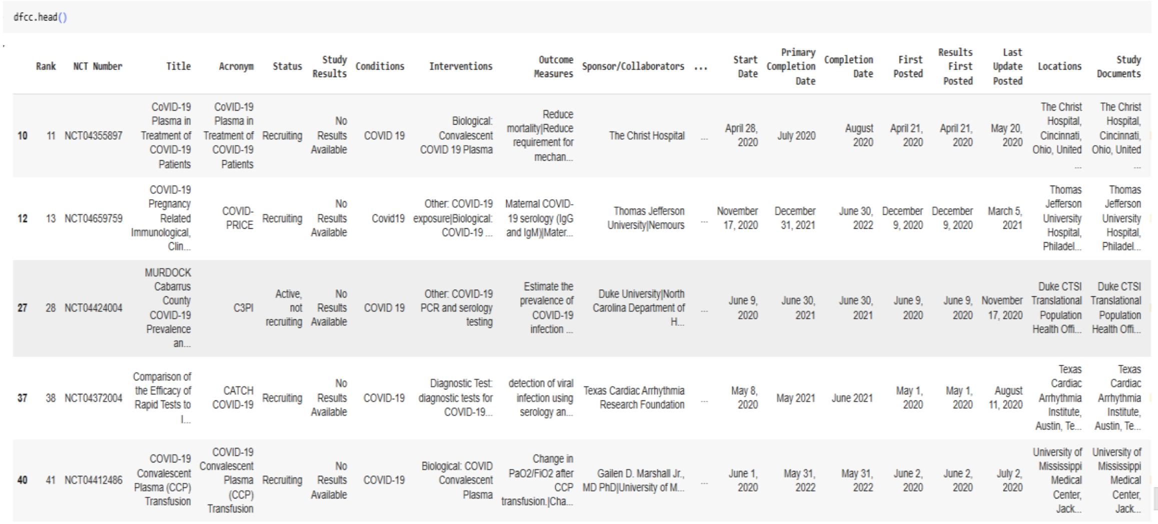
```
dfc.first()
```

	Rank	NCT Number	Title	Acronym	Status	Study Results	Conditions	Interventions	Outcome Measures	Sponsor/Collaborators	...	Other IDs	Start Date	Primary Completion Date	Completion Date	
country																
2020	36	NCT04645498	COVID-19 and Hereditary Metabolic Diseases	COVID19-MHM	Not yet recruiting	No Results Available	Covid19|Metabolism, Inborn Errors	Covid19|Metabolism, Inborn Errors	Frequency of MHM imbalance triggered by COVID-...	University Hospital, Lille	...	2020_52|2020-A02886-33	January 2021	January 2023	January 2023	No 2
2021	46	NCT04834908	Evaluation of Equine Antibody Treatment in Pat...	PROTECT	Not yet recruiting	No Results Available	SARS-CoV-2 Infection	Biological: Equine COVID-19 Antiserum|Drug: St...	Phase 1 Unexpected serious adverse events|Phas...	Bharat Serums and Vaccines Limited	...	BSV_EQ-AB_20_08	May 2021	March 2022	September 2022	
Africa	423	NCT04709302	Effects of COVID-19 on Endothelium in HIV-Posi...	ENDOCOVID	Recruiting	No Results Available	Covid19|Hiv|ART	Biological: COVID-19|Biological: HIV|Drug: ART	Number of patients developing Acute Respirator...	Medical University of Graz|University of Olso|...	...	ENDOCOVID	February 2021	June 2022	December 2022	1
Albania	73	NCT04811391	COVID-19 Vaccine Effectiveness in Albanian Hea...	COVEAL	Recruiting	No Results Available	Covid19	Biological: COVID-19 vaccine	COVID-19 vaccine effectiveness|COVID-19 PCR co...	Institute of Public Health, Albania|World Heal...	...	COV-VE-0001	February 19, 2021	May 20, 2022	May 20, 2022	Ma

- Shows the first row for each country group

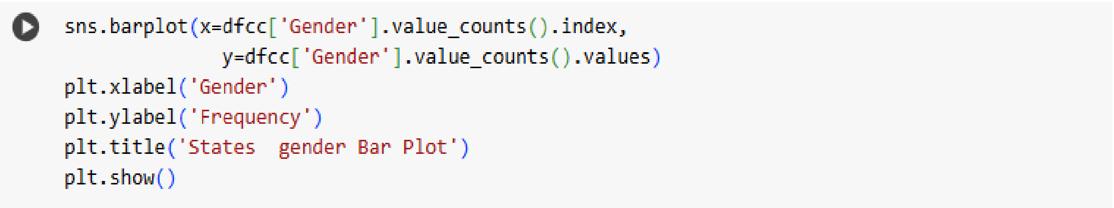
```
dfcc=dfc.get_group('States')
```

- dfcc is a new dataframe that contains only studies from "States".

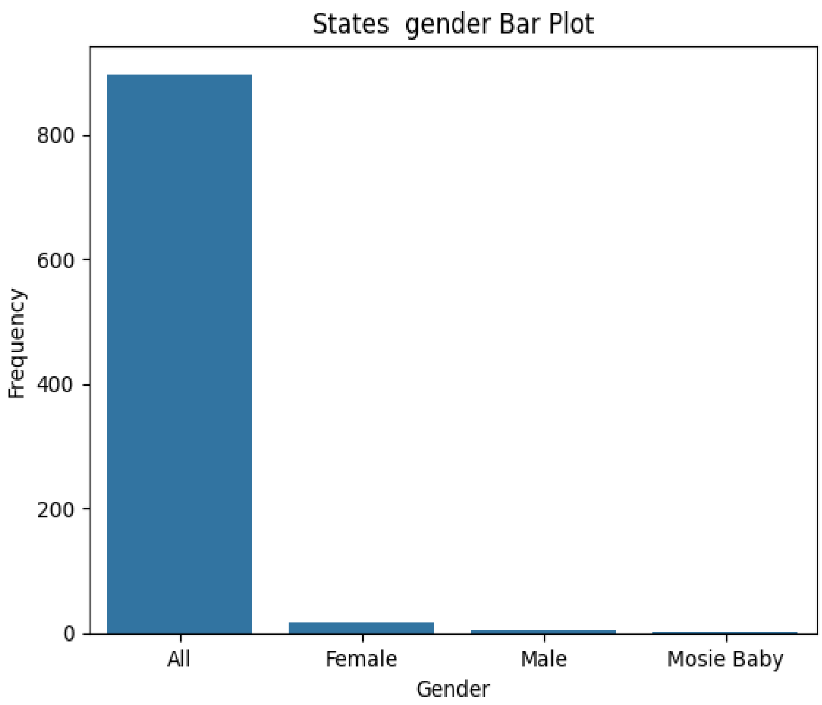


```
dfcc.head()
```

	Rank	NCT Number	Title	Acronym	Status	Study Results	Conditions	Interventions	Outcome Measures	Sponsor/Collaborators	...	Start Date	Primary Completion Date	Completion Date	First Posted	Results First Posted	Last Update Posted	Locations	Study Documents
10	11	NCT04355897	CoVID-19 Plasma in Treatment of COVID-19 Patients	CoVID-19 Plasma in Treatment of COVID-19 Patients	Recruiting	No Results Available	COVID 19	Biological: Convalescent COVID 19 Plasma	Reduce mortality|Reduce requirement for mechan...	The Christ Hospital	...	April 28, 2020	July 2020	August 2020	April 21, 2020	April 21, 2020	May 20, 2020	The Christ Hospital, Cincinnati, Ohio, United ...	The Christ Hospital, Cincinnati, Ohio, United ...
12	13	NCT04659759	COVID-19 Pregnancy Related Immunological, Clin...	COVID-PRICE	Recruiting	No Results Available	Covid19	Other: COVID-19 exposure|Biological: COVID-19 ...	Maternal COVID-19 serology (IgG and IgM)|Mater...	Thomas Jefferson University|Nemours	...	November 17, 2020	December 31, 2021	June 30, 2022	December 9, 2020	December 9, 2020	March 5, 2021	Thomas Jefferson University Hospital, Philadel...	Thomas Jefferson University Hospital, Philadel...
27	28	NCT04424004	MURDOCK Cabarrus County COVID-19 Prevalence an...	C3PI	Active, not recruiting	No Results Available	COVID 19	Other: COVID-19 PCR and serology testing	Estimate the prevalence of COVID-19 infection ...	Duke University|North Carolina Department of H...	...	June 9, 2020	June 30, 2021	June 30, 2021	June 9, 2020	June 9, 2020	November 17, 2020	Duke CTSI Translational Population Health Offi...	Duke CTSI Translational Population Health Offi...
37	38	NCT04372004	Comparison of the Efficacy of Rapid Tests to I...	CATCH COVID-19	Recruiting	No Results Available	COVID-19	Diagnostic Test: diagnostic tests for COVID-19...	detection of viral infection using serology an...	Texas Cardiac Arrhythmia Research Foundation	...	May 8, 2020	May 2021	June 2021	May 1, 2020	May 1, 2020	August 11, 2020	Texas Cardiac Arrhythmia Institute, Austin, Te...	Texas Cardiac Arrhythmia Institute, Austin, Te...
40	41	NCT04412486	COVID-19 Convalescent Plasma (CCP) Transfusion	COVID-19 Convalescent Plasma (CCP) Transfusion	Recruiting	No Results Available	COVID-19	Biological: COVID Convalescent Plasma	Change in PaO2/FiO2 after CCP transfusion.|Cha...	Gailen D. Marshall Jr., MD PhD|University of M...	...	June 1, 2020	May 31, 2022	May 31, 2022	June 2, 2020	June 2, 2020	July 2, 2020	University of Mississippi Medical Center, Jack...	University of Mississippi Medical Center, Jack...



```
sns.barplot(x=dfcc['Gender'].value_counts().index,
              y=dfcc['Gender'].value_counts().values)
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.title('States  gender Bar Plot')
plt.show()
```



5. Conclusion

1. Top Countries for Clinical Trials

- The **United States (States)** conducted the maximum number of COVID-19 clinical trials.
- **France**, **United Kingdom**, **Italy**, and **Spain** were also major contributors.
- This shows that developed countries were leading clinical research during the pandemic.

2. Funding Sources

- Most clinical trials were funded under the category "**Other**".
- The second largest funding source was "**Industry**" (pharmaceutical companies, biotech firms).
- Combination funding like **Industry|Other** also appeared but was much less frequent.

- **Conclusion**: Clinical trials were majorly **privately or independently funded** rather than solely by governments or official health bodies.

3. **Gender Distribution in Studies (for 'States')**

- A majority of clinical trials were open to **All Genders** (both male and female participants).
- Some studies focused specifically on **Male** or **Female** participants, but these were much fewer.
- **Conclusion**: Researchers mostly designed studies to include **both genders**, aiming for a broader understanding of COVID-19's impact.

4. **Timeline Observations (if any)**

- A lot of trials spiked around **2020-2021**, which aligns perfectly with the global emergency and vaccine development phases.