



Project Title	COVID-19 Clinical Trials EDA Pandas
Tools	Python, ML, SQL, Excel
Technologies	Data Analyst & Data scientist
Project Difficulties level	intermediate

Dataset : Dataset is available in the given link. You can download it at your convenience.

[Click here to download data set](#)

About Dataset

Dataset Description

ClinicalTrials.gov is a database of privately and publicly funded clinical studies conducted around the world. It is maintained by the National Institute of Health. All data is publicly available and the site provides a direct download feature which makes it super easy to use relevant data for analysis.

This dataset consists of clinical trials related to COVID 19 studies presented on the site.

The dataset consists of XML files where each XML file corresponds to one study. The filename is the NCT number which is a unique identifier of a study in the ClinicalTrials repository. Additionally, a CSV file has also been provided, which might not have as much information as contained in the XML file, but does give sufficient information.

Please refer to this notebook for details on the dataset :
<https://www.kaggle.com/parulpandey/eda-on-covid-19-clinical-trials>

Acknowledgements

ClinicalTrials.gov is a resource provided by the U.S. National Library of Medicine.

IMPORTANT:

Listing a study does not mean it has been evaluated by the U.S. Federal Government. Read our disclaimer for details.

Before participating in a study, talk to your health care provider and learn about the risks and potential benefits.

NOTE :

1. this project is only for your guidance, not exactly the same you have to create. Here I am trying to show the way or idea of what steps you can follow and how your projects look. Some projects are very advanced (because it will be made with the help of flask, nlp, advance ai, advance DL and some advanced things) which you can not understand .
2. You can make or analyze your project with yourself, with your idea, make it more creative from where we can get some information and understand about our business. make sure what overall things you have created all things you understand very well.

Example

what steps you should have to follow

Here's a step-by-step guide for performing Exploratory Data Analysis (EDA) on a COVID-19 Clinical Trials dataset using Pandas, tailored for beginners.

Project Title:

Exploratory Data Analysis of COVID-19 Clinical Trials

1. Objective

The objective is to explore the dataset to gain insights into the characteristics of

COVID-19 clinical trials, such as their status, phases, study designs, and demographics.

2. Importing Libraries and Loading Data

First, you'll need to import the necessary libraries and load your dataset.

```
import pandas as pd

# Load the dataset
df = pd.read_csv('covid_clinical_trials.csv') # Replace with
your dataset's path
```

3. Initial Data Exploration

Start by exploring the basic structure and content of the dataset.

```
# View the first few rows of the dataset
print(df.head())

# Check the columns and data types
print(df.info())

# Summary statistics for numerical columns
print(df.describe())

# Summary statistics for categorical columns
print(df.describe(include='object'))
```

4. Handling Missing Data

Check for missing values and decide how to handle them.

```
# Check for missing values
print(df.isnull().sum())

# Drop columns with a high percentage of missing values or fill
them
```

```
df = df.drop(columns=['Acronym', 'Study Documents']) # Example
of dropping columns
df['Results First Posted'].fillna('Unknown', inplace=True) #
Example of filling missing data
```

5. Univariate Analysis

Analyze each column individually to understand the distribution and key characteristics.

- **Status Distribution:** Analyze the status of clinical trials (e.g., Completed, Ongoing).

```
print(df['Status'].value_counts())
df['Status'].value_counts().plot(kind='bar', title='Status of
Clinical Trials')
```

- **Phase Distribution:** Understand the distribution of trial phases.

```
print(df['Phases'].value_counts())
df['Phases'].value_counts().plot(kind='bar',
title='Distribution of Phases')
```

- **Age Group Analysis:** Analyze the distribution of age groups.

```
print(df['Age'].value_counts())
df['Age'].value_counts().plot(kind='bar', title='Age Group
Distribution')
```

6. Bivariate Analysis

Explore relationships between different variables.

- **Status vs. Phases:** Explore how trial phases are distributed across different statuses.

```
status_phase = pd.crosstab(df['Status'], df['Phases'])
print(status_phase)
status_phase.plot(kind='bar', stacked=True, title='Status vs.
Phases')
```

- **Conditions vs. Outcome Measures:** Understand the common outcome measures for different conditions.

```
conditions_outcomes = df.groupby('Conditions')['Outcome Measures'].apply(lambda x: ', '.join(x)).reset_index()
print(conditions_outcomes)
```

7. Time Series Analysis

Analyze the trends over time, such as the number of trials started over the months.

```
# Convert date columns to datetime
df['Start Date'] = pd.to_datetime(df['Start Date'],
errors='coerce')
df['Primary Completion Date'] = pd.to_datetime(df['Primary Completion Date'], errors='coerce')

# Plot the number of trials started over time
df['Start Date'].dt.to_period('M').value_counts().sort_index().plot(kind='line', title='Trials Started Over Time')
```

8. Conclusion

Summarize the findings from your EDA. For example:

- The majority of trials are in the "Completed" phase.
- Most trials target adult populations.
- There's a steady increase in the number of trials over time.

9. Saving Results

You can save the processed data or specific analysis results for further use.

```
# Save the cleaned data
df.to_csv('cleaned_covid_clinical_trials.csv', index=False)
```

10. Output and Visuals

After running the code, you should observe:

- Bar charts showing the distribution of trial statuses, phases, and age groups.
- A time series plot illustrating the trend of trials over time.

This project will provide a solid foundation in EDA using Pandas, with practical insights into the clinical trials landscape for COVID-19.

Sample code

Import Required Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Load The DataSet

In [2]:

```
df = pd.read_csv('../input/covid19-clinical-trials-dataset/COVID clinical
trials.csv' , index_col = 0)
```

Exploratory Data Analysis

In [3]:

```
# print the first 5 rows in the dataset
df.head(n = 5)
```

Out[3]:

	NCT Number	Title	Acronym	Status	Study Results	Conditions	Interventions	Outcome Measures	Sponsor/Collaborators	Gender	Other IDs	Start Date	Primary Completion Date	Completion Date	First Posted	Results First Posted	Last Updated Posted	Locations	Study Documents	URL
Rank																				
1	NCT04785898	Diagnostic Performance of the ID Now™ COVID-19...	COVID-ID Now	Active, not recruiting	No Results Available	Covid 19	Diagnostic Test: ID Now™ COVID-19 Screening Test	Evaluate the diagnostic performance of the ID ...	Group e Hospitalier Paris Saint Joseph	All	COVID-ID Now	November 9, 2020	December 22, 2020	April 30, 2021	March 8, 2021	March 8, 2021	March 8, 2021	Gr oupe Hospitalier Paris Saint-Joseph, Paris , ...	N a N	https://ClinicalTrials.gov/show/NCT04785898

2	NC T04595136	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	AILI	...	COVID19-0001-USR	November 2, 2020	December 15, 2020	January 29, 2021	October 20, 2020	NAN	October 20, 2020	Cimedic al, Barra nguilla, Atlantico, Colombia	NAN	https://ClinicalTrials.gov/show/NCT04595136
3	NC T04395482	Lung CT Scan Analysis of SARS-CoV2 Induced Lung...	TAC-COVID19	Recruiting	No Results Available	covid 19	Other: Lung CT scan analysis in COVID-19 patients	A qualitative analysis of parenchymal lung dam..	University of Milano Bicocca	AILI	...	TAC-COVID19	May 7, 2020	June 15, 2021	June 15, 2021	May 20, 2020	NAN	November 9, 2020	Ospedale Paganini XII, Bergamo, Italy P..	NAN	https://ClinicalTrials.gov/show/NCT04395482

4	NC T0 44 16 06 1	The Ro le of a Pri vat e Ho spi tal in Ho ng Ko ng A m...	C O V I D - 1 9	A ct ive , n o t r e c r ui t i n g	N o R e s u l t s A v a i l a b l e	COVI D	Di ag no sti c T e s t: C O V I D 19 Di ag no sti c T e s t	Prop or tion of asym ptom atic subje cts Pr oport ion...	Hong Kong Sanat orium & Hospi tal	A l l	.	R C -2 0 2 0- 0 8	M a y 2 5, 2 0 2 0	Ju ly 3 1, 2 0 2 0	A u g u s t 3 1, 2 0 2 0	J u n e 4 , 2 0 2 0	N a N	J u n e 4, 2 0 2 0	H o n g K o n g S a n a t o r i u m & H o s p i t a l, H o n g K o n g , H o...	N a N	https://ClinicalTrials.gov/show/NCT04416061
5	NC T0 43 95 92 4	Ma ter nal -fo eta l Tr an s m i s s i o n of S A R S -C o v -2	T M F - C O V I D - 1 9	R e c r ui t i n g	N o R e s u l t s A v a i l a b l e	Mat er nal F e t al I n f e c t i o n T r a n s m i s s i o n C O V I D -19...	Di ag no sti c T e s t: Di ag n o s i s o f S A R S -C o v 2 b y R T -...	COVI D-19 by posi tive PCR in cord blood and / o...	Cent r e H o s p i t a l i e r R é g i o n al d' O r lé a n s C e n t r e d...	F e m a l e	.	C H R O -2 0 2 0- 0 1 0	M a y 5, 2 0 2 0	M a y 2 0 2 1	M a y 2 0 2 1	M a y 2 0 , 2 0 2 0	N a N	J u n e 4, 2 0 2 0	C H R O r lé a n s, O r lé a n s, F r a n c e	N a N	https://ClinicalTrials.gov/show/NCT04395924

5 rows × 26 columns

In [4]:

```
# Shape of the DataSet
df.shape
```

Out[4]:

```
(5783, 26)
```

In [5]:

```
# Columns in the dataset
df.columns
```

Out[5]:

```
Index(['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')
```

In [6]:

```
# Categorical Features
df.select_dtypes(include = 'object').columns
```

Out[6]:

```
Index(['NCT Number', 'Title', 'Acronym', 'Status', 'Study Results',
      'Conditions', 'Interventions', 'Outcome Measures',
      'Sponsor/Collaborators', 'Gender', 'Age', 'Phases', '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')
```

In [7]:

```
# Neumrical Features
df.select_dtypes(exclude = 'object').columns
```

Out[7]:

```
Index(['Enrollment'], dtype='object')
```

In [8]:

```
# Detecting (Percentage) Missing Data
missing_data = df.isnull().mean() * 100
missing_data
```

Out[8]:

NCT Number	0.000000
Title	0.000000
Acronym	57.115684
Status	0.000000
Study Results	0.000000
Conditions	0.000000
Interventions	15.320768
Outcome Measures	0.605222
Sponsor/Collaborators	0.000000
Gender	0.172921
Age	0.000000
Phases	42.555767
Enrollment	0.587930
Funded Bys	0.000000
Study Type	0.000000
Study Designs	0.605222
Other IDs	0.017292
Start Date	0.587930
Primary Completion Date	0.622514
Completion Date	0.622514
First Posted	0.000000
Results First Posted	99.377486
Last Update Posted	0.000000
Locations	10.115857
Study Documents	96.852845
URL	0.000000

dtype: float64

In [9]:

```
# Visualize data without calculating
```

```
def visualize_data(data , caption = '' , ylabel = 'Percentage of Missing Data'):
```

```
    # set figure size
```

```
    sns.set(rc={'figure.figsize':(15,8.27)})
```

```
    # make ticks vertical
```

```
    plt.xticks(rotation=90)
```

```
    # set title to the image and plot it or the highest 40
```

```
    fig = sns.barplot(x = data.keys()[ :min(40 , len(data))].tolist() , y =  
data.values[ : min(40 , len(data))].tolist() ) \  
    .set_title(caption)
```

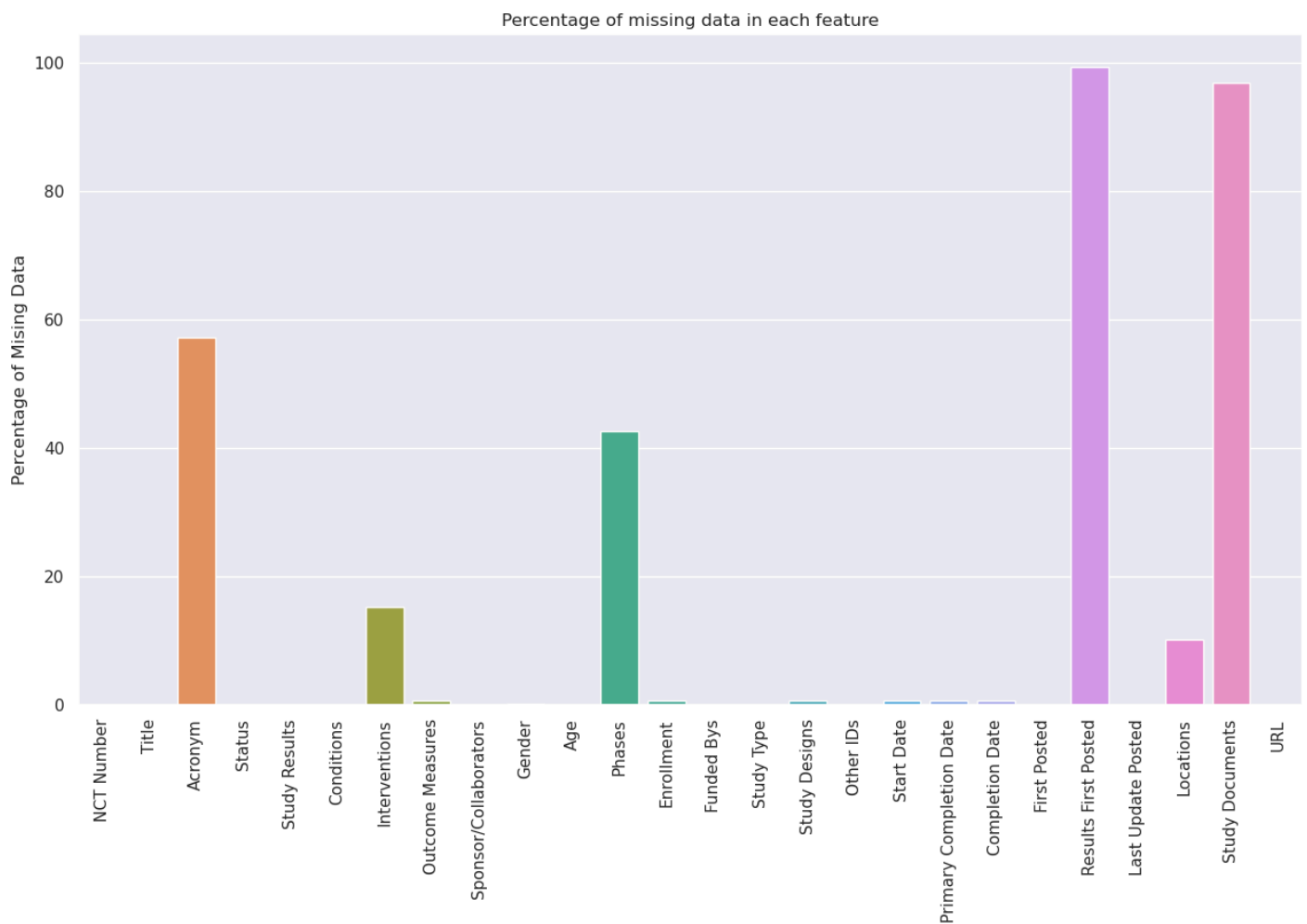
```
    # set labels
```

```
    plt.ylabel(ylabel)
```

```
    plt.show()
```

In [10]:

```
visualize_data(missing_data , 'Percentage of missing data in each feature')
```



As shown the percentae of missing data in **Results First Posted** is **99.3%** and **Study Documents** is **96.8%**, so it's impossible to impute them without destroying our dataset.

In [11]:

```
# Drop Study Documents and Results First Posted
df.drop(['Results First Posted' , 'Study Documents'] , inplace = True , axis = 1 )
```

In [12]:

```
# Columns in the dataset after dropping Study Documents and Results First Posted
df.columns
```

Out[12]:

```
Index(['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',  
'Last Update Posted', 'Locations', 'URL'],
```

```
dtype='object')
```

In [13]:

```
# Drop Duplicate Rows  
print(f"Shape before dropping duplicates data {df.shape}")  
df.drop_duplicates(inplace = True)  
print(f"Shape after dropping duplicates data {df.shape}")
```

```
Shape before dropping duplicates data (5783, 24)  
Shape after dropping duplicates data (5783, 24)
```

There is no duplicate rows in the dataset.

In [14]:

```
# Drop rows that have less than 10 non-null values  
print(f"Shape before dropping Null rows {df.shape}")  
df.dropna(how = 'any' , axis = 0 , thresh = 10 , inplace = True)  
print(f"Shape after dropping Null rows {df.shape}")
```

```
Shape before dropping Null rows (5783, 24)  
Shape after dropping Null rows (5783, 24)
```

There is no rows with less than 10 non-null values

In [15]:

```
df.isnull().mean() * 100
```

Out[15]:

```
NCT Number          0.000000  
Title               0.000000  
Acronym             57.115684
```

```
Status          0.000000
Study Results    0.000000
Conditions       0.000000
Interventions    15.320768
Outcome Measures 0.605222
Sponsor/Collaborators 0.000000
Gender          0.172921
Age             0.000000
Phases          42.555767
Enrollment      0.587930
Funded Bys      0.000000
Study Type      0.000000
Study Designs    0.605222
Other IDs       0.017292
Start Date      0.587930
Primary Completion Date 0.622514
Completion Date 0.622514
First Posted    0.000000
Last Update Posted 0.000000
Locations       10.115857
URL             0.000000
```

```
dtype: float64
```

In [16]:

```
# We can extract a new feature form The Location which is the country where the study hold
countries = [ str(df.Locations.iloc[i]).split(',')[ -1] for i in range(df.shape[0])]
df['Country'] = countries
```

In [17]:

```
df.columns
```

Out[17]:

```
Index(['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',
      'Last Update Posted', 'Locations', 'URL', 'Country'],
      dtype='object')
```

In [18]:

```
df.Country.value_counts()[:35]
```

Out[18]:

United States	1267
France	647
nan	585
United Kingdom	306
Italy	235
Spain	234
Turkey	219
Canada	202
Egypt	192
China	171
Brazil	137
Germany	128
Belgium	91
Mexico	88
Switzerland	76
Russian Federation	69
Sweden	57
Denmark	56
Israel	56
India	55
Pakistan	53
Argentina	47
Netherlands	46
Norway	38
Hong Kong	36
Colombia	33
Republic of	31
Austria	29
Poland	29
Singapore	29
Saudi Arabia	27
Australia	26
Greece	26
Islamic Republic of	23
South Africa	22

Name: Country, dtype: int64

Now We need to clasify the missing data to one of these categories

1) Missing Completely At Random (MCAR)

2) Missing At Random (MAR)

3) Not Missing At Random (NMAR)

In [19]:

```
# Lets's start with Acronym
```

```
print(f"Number of unique values is {df.Acronym.nunique()} \n")
df.Acronym.value_counts()
```

```
Number of unique values is 2338
```

Out[19]:

COVID-19	47
PROTECT	7
CORONA	6
RECOVER	5
SCOPE	5
..	
ASD	1
VICO	1
LICORNE	1
LOSVID	1
MindMyMindFU	1

```
Name: Acronym, Length: 2338, dtype: int64
```

In [20]:

```
# Find the realtion between null values in Acronym and Countries
(df.Acronym.isnull().groupby(df.Country).mean().sort_values(ascending = False) *
100)[:60]
```

Out[20]:

Country	
Iraq	100.000000
Belarus	100.000000
Rwanda	100.000000
South Sudan	100.000000
Cambodia	100.000000

Bulgaria	100.000000
Cyprus	100.000000
Bosnia and Herzegovina	100.000000
Guinea-Bissau	100.000000
Dominican Republic	100.000000
Ecuador	100.000000
North Macedonia	100.000000
Bahrain	100.000000
Azerbaijan	100.000000
Uruguay	100.000000
Uzbekistan	100.000000
Kyrgyzstan	100.000000
Cape Verde	100.000000
Republic of	96.774194
Taiwan	93.750000
Singapore	93.103448
Japan	88.888889
Kuwait	87.500000
China	87.134503
Turkey	86.757991
Ukraine	85.714286
Malaysia	84.615385
Egypt	83.854167
Hungary	83.333333
Hong Kong	80.555556
Bangladesh	80.000000
India	80.000000
Kazakhstan	80.000000
Saudi Arabia	77.777778
Puerto Rico	76.470588
Israel	75.000000
Zimbabwe	75.000000
Jordan	72.727273
Poland	72.413793
Indonesia	71.428571
United States	69.376480
Romania	69.230769
Kenya	66.666667
Nepal	66.666667
New Zealand	66.666667
Ethiopia	66.666667
Slovakia	66.666667
Thailand	66.666667
Lebanon	66.666667
nan	66.324786
Islamic Republic of	65.217391
Russian Federation	65.217391
Chile	64.705882
Austria	62.068966
Pakistan	60.377358
Brazil	59.124088

Mexico	57.954545
Sweden	57.894737
Argentina	57.446809
Canada	55.940594

Name: Acronym, dtype: float64

- After inspecting the relation between the missing values in Acronym and Country we can conclude that there is a sort of relation between these two features, so we can say that Data is Missing At Random (MAR).
- So we can Impute by Missing Category.

In [21]:

```
# impute by a missing Indicator
df.Acronym = df.Acronym.fillna("Missing Acronym")
```

In [22]:

```
# Detecting (Percentage) Missing Data
df.isnull().mean() * 100
```

Out[22]:

NCT Number	0.000000
Title	0.000000
Acronym	0.000000
Status	0.000000
Study Results	0.000000
Conditions	0.000000
Interventions	15.320768
Outcome Measures	0.605222
Sponsor/Collaborators	0.000000
Gender	0.172921
Age	0.000000
Phases	42.555767
Enrollment	0.587930
Funded Bys	0.000000
Study Type	0.000000
Study Designs	0.605222
Other IDs	0.017292
Start Date	0.587930
Primary Completion Date	0.622514
Completion Date	0.622514
First Posted	0.000000
Last Update Posted	0.000000
Locations	10.115857

```
URL                0.000000
Country            0.000000
```

```
dtype: float64
```

We can do the same for other categorical features such as Interventions , Phases , Locations and other categorical features

In [23]:

```
# Impute Interventions , Phases , Locations by Missing Category

categorical_features = df.select_dtypes(include = object).columns

features = categorical_features[df[categorical_features].isnull().mean() > 0]

for feature in features:
    df[feature] = df[feature].fillna(f"Missing {feature}")
```

In [24]:

```
# Detecting (Percentage) Missing Data
df.isnull().mean() * 100
```

Out[24]:

```
NCT Number        0.00000
Title              0.00000
Acronym            0.00000
Status             0.00000
Study Results      0.00000
Conditions          0.00000
Interventions      0.00000
Outcome Measures   0.00000
Sponsor/Collaborators 0.00000
Gender             0.00000
Age                0.00000
Phases             0.00000
Enrollment         0.58793
Funded Bys         0.00000
Study Type         0.00000
Study Designs      0.00000
Other IDs          0.00000
Start Date         0.00000
Primary Completion Date 0.00000
Completion Date    0.00000
```

```
First Posted      0.00000
Last Update Posted 0.00000
Locations         0.00000
URL               0.00000
Country           0.00000
```

```
dtype: float64
```

Now the Time to handle The missing data for the Enrollment

In [25]:

```
# Check the skewness
df.Enrollment.skew()
```

Out[25]:

```
34.06593382031148
```

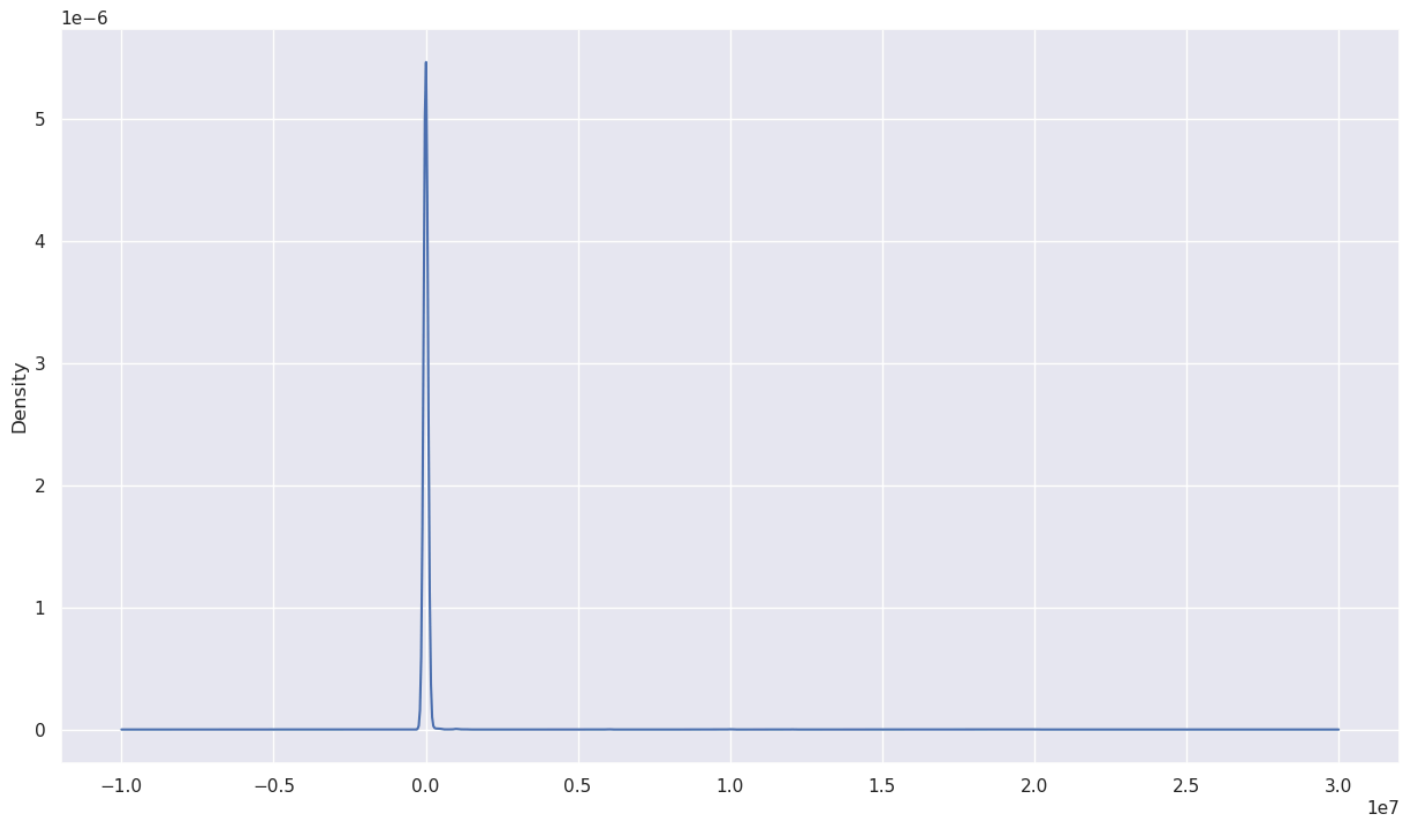
The value of Skewness is 34 which means that we This feature isn't normally distributed

In [26]:

```
# Plotting the distribution of the enrollment
df.Enrollment.plot(kind = 'kde')
```

Out[26]:

```
<AxesSubplot:ylabel='Density'>
```



So We will impute by the median

In [27]:

```
# Some Statistical Value for the Enrollment Column
```

```
min_Value = df.Enrollment.min()
max_Value = df.Enrollment.max()
mean_Value = df.Enrollment.mean()
median_Value = df.Enrollment.median()
std_Value = df.Enrollment.std()

print(f"the min value is {min_Value} \n \
The max value is {max_Value} \n \
The mean is {mean_Value} \n \
The Median is {median_Value} \n \
Standard Deviation is {std_Value}")
```

```
the min value is 0.0
The max value is 20000000.0
The mean is 18319.48860671421
The Median is 170.0
Standard Deviation is 404543.7287841079
```

In [28]:

```
# Using Median to impute Missing Values
df.Enrollment = df.Enrollment.fillna(median_Value)
```

In [29]:

```
# Detecting (Percentage) Missing Data
df.isnull().mean() * 100
```

Out[29]:

NCT Number	0.0
Title	0.0
Acronym	0.0
Status	0.0
Study Results	0.0
Conditions	0.0
Interventions	0.0
Outcome Measures	0.0
Sponsor/Collaborators	0.0
Gender	0.0
Age	0.0
Phases	0.0
Enrollment	0.0
Funded Bys	0.0
Study Type	0.0
Study Designs	0.0
Other IDs	0.0
Start Date	0.0
Primary Completion Date	0.0
Completion Date	0.0
First Posted	0.0
Last Update Posted	0.0
Locations	0.0
URL	0.0
Country	0.0

dtype: float64

In [30]:

```
df.head()
```

Out[30]:

	NC T Nu mb er	Tit le	A c r o n y m	S t a t u s	S t u d y R e s u l t s	Con d i t i o n s	Int er v e n t i o n s	Out c o m e M e a s u r e s	Spon sor/C ollab o r a t o r s	G e n d e r	Study Desig ns	O t h e r I D s	S t a r t D a t e	Pr i m a r y C o m p l e t i o n D a t e	C o m p l e t i o n D a t e	F i r s t P o s t e d	L a s t U p d a t e P o s t e d	Lo c a t i o n s	URL	C o u n t r y
	R a n k																			
1	NC T0 47 85 89 8	Di a g n o s t i c P e r f o r m a n c e o f t h e ID N o w™ C O V I D-	C O V I D-I D N o w	A c t i v e , n o t r e c r u i t i n g	N o R e s u l t s A v a i l a b l e	Covid 19	Di a g n o s t i c T e s t: ID N o w™ C O V I D-19 S c r e e n i n g	E v a l u a t e t h e diag n o s t i c p e r f o r m a n c e o f t h e ID ...	Grou pe Hospi talier Paris Saint Jose ph	A l l o c a t i o n : N/A Int er v e n t i o n M o d e l: Single Gro...	C O V I D-I D N o w	N o v e m b e r 9 , 2 0 2 0	D e c e m b e r 2 2, 2 0 2 0	A p r i l 3 0, 2 0 2 1	M a r c h 8 , 2 0 2 1	M a r c h 8 , 2 0 2 1	Grou pe Hospi talier Paris Saint -J os e p h, P a r i s,	https://Clinical Trials.gov/show/NCT04785 898	F r a n c e	

		19 ...					Test										...			
2	NC T04595136	Study to Evaluate the Efficacy of COVID-19-001..	COVID-19	Not yet recruiting	NO Results Available	SARS-CoV-2 Infection	Drug : Drug COVID-19-001-USR Drug : normal saline	Change on viral load results from baseline aft...	United Medical Specialties	A I I . . .	Allocation: Randomized Intervention Model: Par...	COVID 19 - 00001 - USR	November 2, 2020	December 15, 2020	January 29, 2021	October 20, 2020	October 20, 2020	Cimedic al, Baranquilla, Atlantico, Colombia	https://ClinicalTrials.gov/show/NCT04595136	Colombia
3	NC T04395482	Lung CT Scan Analysis of SARS-CoV2 Induced Lu	TAC - COVID 19	Recruiting	NO Results Available	covid 19	Other: Lung CT scan analysis in COVID-19 patient	A qualitative analysis of parenchymal lung dam. ..	Unive rsity of Milano Bicoc ca	A I I . . .	Observation al Model: Cohort Time Persp ective: ...	TAC - COVID 19	May 7, 2020	June 15, 2021	June 15, 2021	May 20, 2020	November 9, 2020	O sped ale P apa Gi ov an ni X XI II, Ber ga mo, Ita ly P..	https://ClinicalTrials.gov/show/NCT04395482	San Marino

4	NC T04416061	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	Al.	Observational Model: Cohort Time Perspective: ...	RC - 2020 - 08	May 25, 2020	July 31, 2020	August 31, 2020	June 4, 2020	June 4, 2020	Hong Kong Sanatorium & Hospital, Hong Kong, Ho..	https://ClinicalTrials.gov/show/NCT04416061	Hong Kong
5	NC T04395924	Maternal-fetal Transmission of SARS-CoV	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...	Female	Observational Model: Cohort Time Perspective: ...	CHRO - 2020 - 10	May 5, 2020	May 2021	May 2021	May 20, 2020	June 4, 2020	CHROléans, Orléans, France	https://ClinicalTrials.gov/show/NCT04395924	France

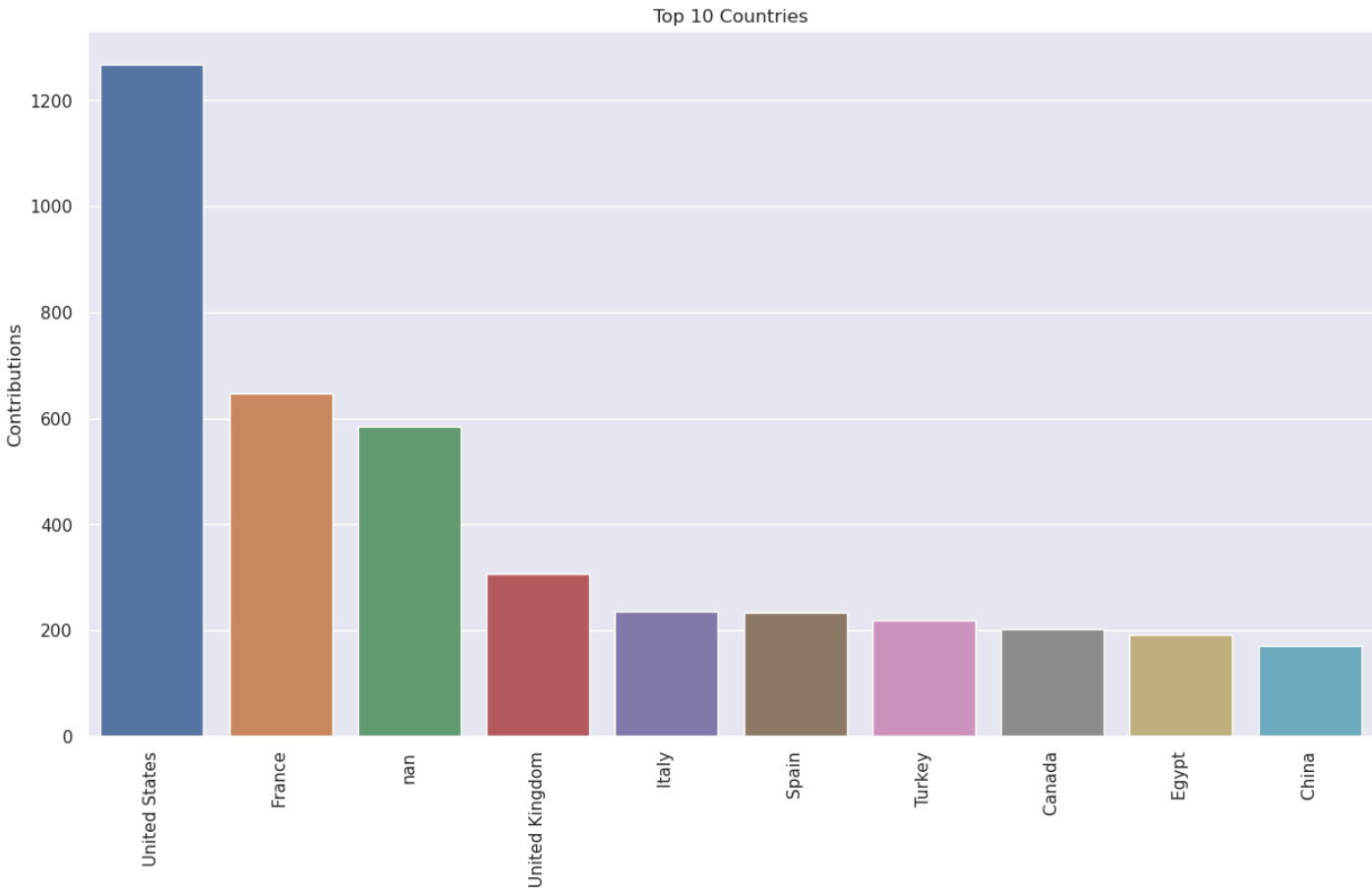
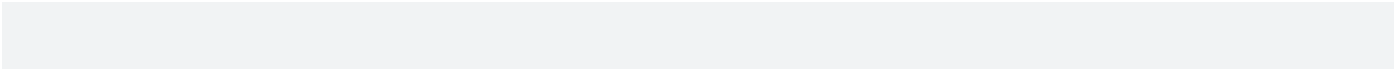
		-2					..																	
--	--	----	--	--	--	--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

5 rows × 25 columns

Data Visualizations

In [31]:

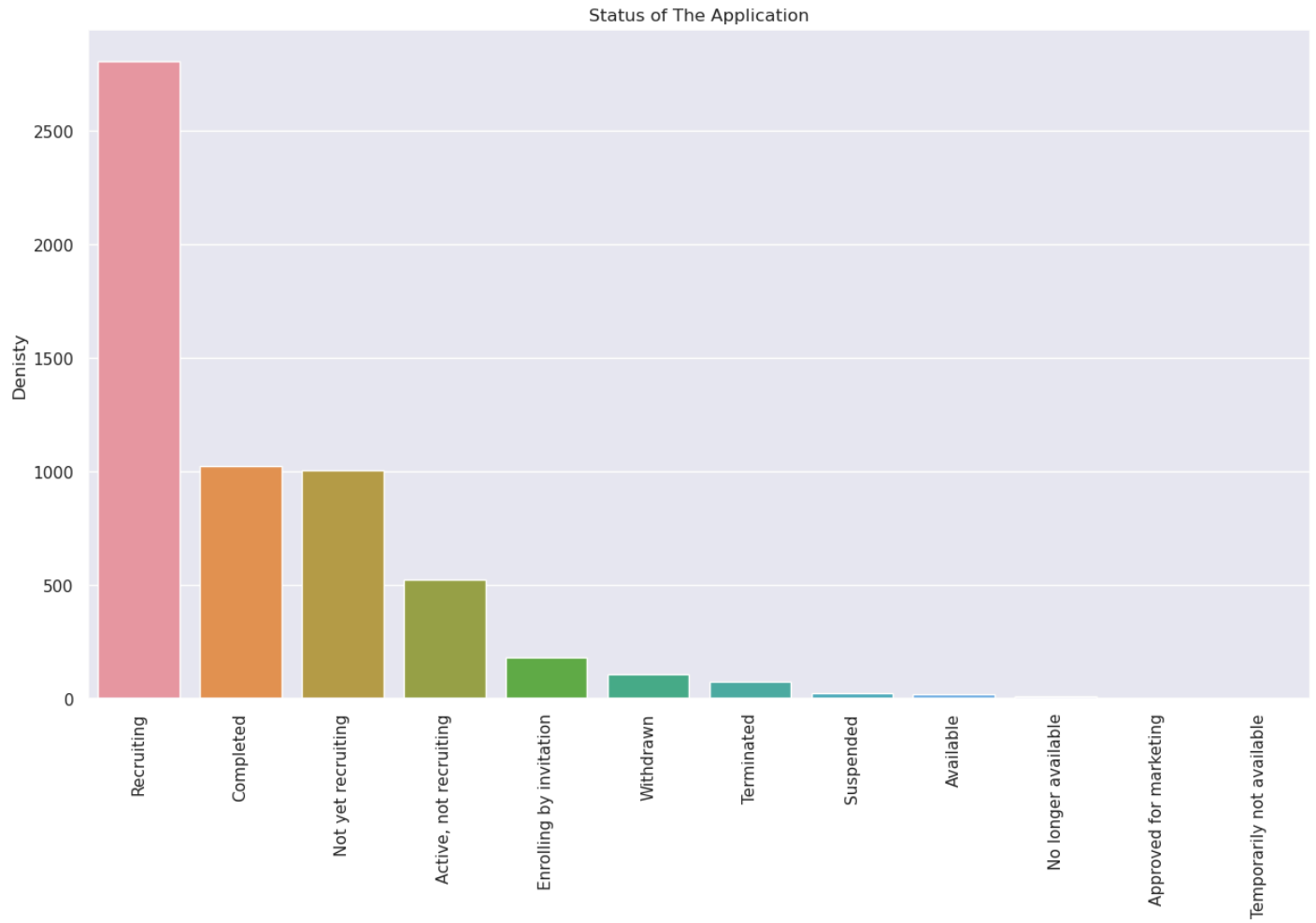
```
# Get Countires with highest Contributiouns
top_10_Countires = df.Country.value_counts()[:10]
visualize_data(top_10_Countires , caption = 'Top 10 Countries' , ylabel =
'Contributions')
```



In [32]:

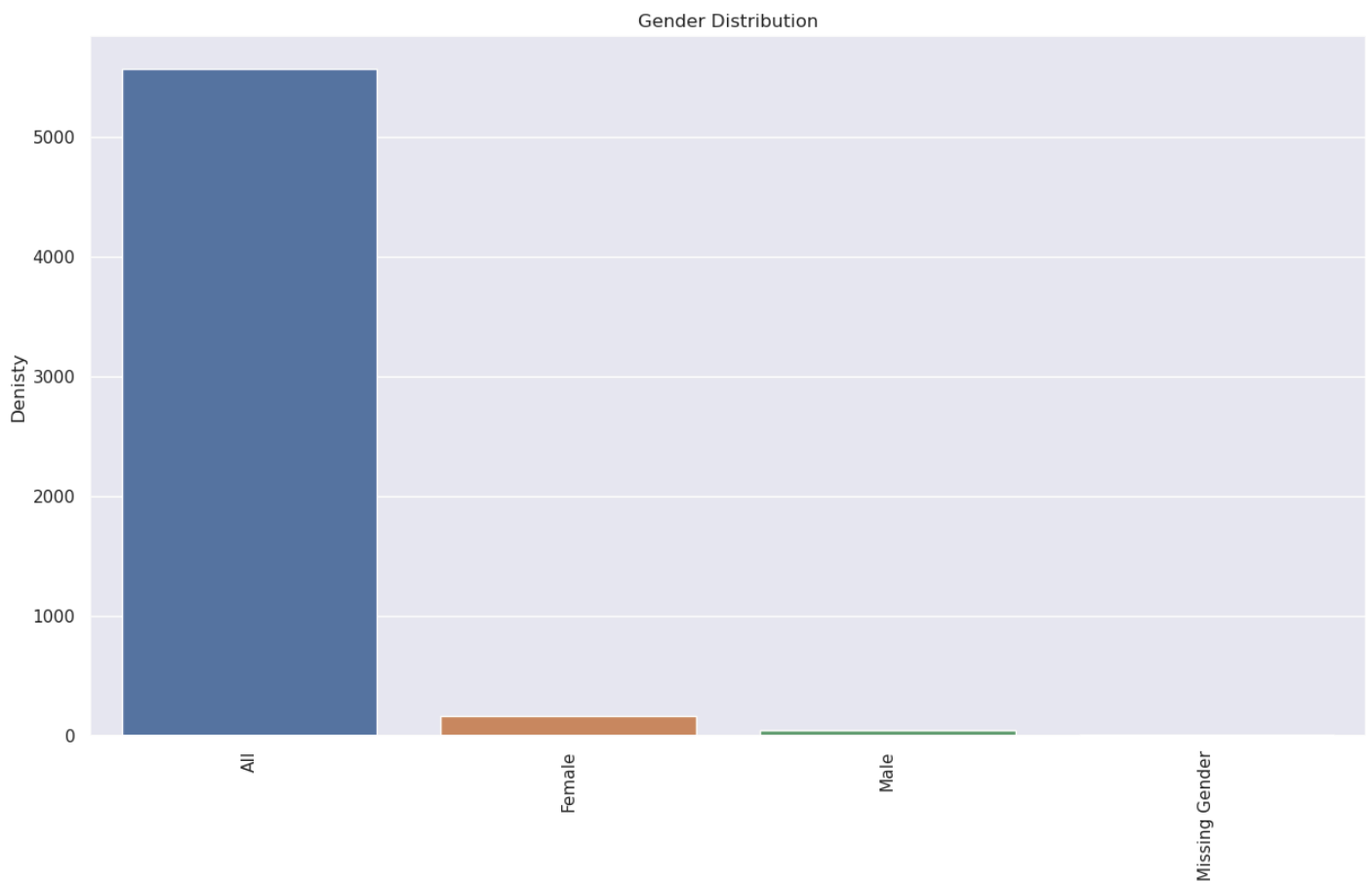
```
# Status of the Application
status = df.Status.value_counts()

visualize_data(status , caption = 'Status of The Application' , ylabel = 'Denisty')
```



In [33]:

```
# Gender Visualiztions
gender = df.Gender.value_counts()
visualize_data(gender , caption = 'Gender Distribution' , ylabel = 'Denisty')
```

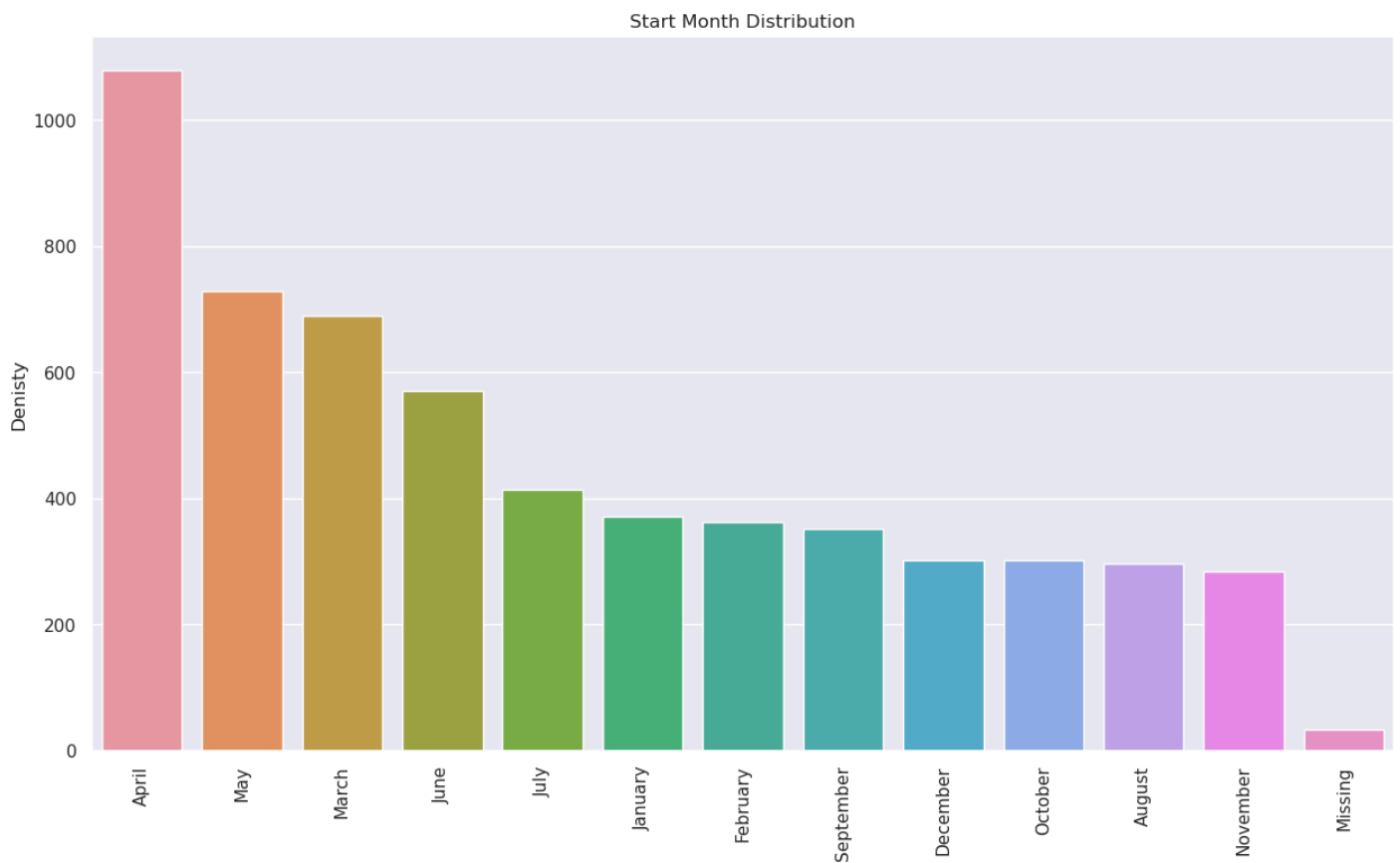


In [34]:

```
# Which month has the highest start
start_month = pd.Series([ str(df['Start Date'].iloc[i]).split(' ')[0] for i in range
(df.shape[0])])

start_month_Distribution = start_month.value_counts()

visualize_data(start_month_Distribution , caption = 'Start Month Distribution' ,
ylabel = 'Density')
```



In [35]:

```
print(f"The shape of data frame is {df.shape}")  
print(f"Nunique in NCT Number is {df['NCT Number'].nunique()}")  
print(f"Nunique in URL is {df.URL.nunique()}")
```

```
The shape of data frame is (5783, 25)  
Nunique in NCT Number is 5783  
Nunique in URL is 5783
```

So If We are going to apply a (Machine Learning) ML model we can drop NCT Number and URL because there is an index already which is Rank. To reduce the number of categorical Features, Specially because they will need to be doecoded inorder to be used in a ML Model.

1 [Reference link](#)

2 [Reference link](#) for ML project