```
import pandas as pd
import numpy as np


#import data from drive

viral_load = pd.read_csv('/content/drive/My Drive/HIV - Machine learning/Datasets/viral_load_dataset.csv')

# display viral load data
viral_load
```

	view_sentinel_events_id	status_date	patient_clinic_no	gender	date_of_birth	diagnosis_date	art_enrollment_dat
0	bb2a12c5-b1ab-4e02-9293-eafc7a51c0a7	4/5/24	16253	female	1/1/94	NaN	Na
1	c9e36b69-21aa-4eb8-9c50-d368bebc1796	4/5/24	621	female	12/1/08	8/12/09	8/12/0
2	e3a74b2d-bcb8-4b8c-b82e-5303ebd6fa7c	4/5/24	DEMO1940	female	12/16/82	8/26/10	9/21/1
3	8cbb3f35-20c8-4a28-abe3-5e995a640920	4/5/24	37	female	1/1/63	8/9/11	8/9/1
4	fb2b022a-3f27-4089-a0d1-abf11548ba14	4/5/24	ACH0700	female	4/12/82	9/4/08	9/4/0
...	...	...	...	...	...	...	.
340359	0c1b8869-8864-4c1c-97b5-98101b8e5168	4/5/24	DWL0610KM	female	5/25/90	4/14/15	4/14/1
340360	e3109b7d-368d-47c4-aea6-dcefa4561ba4	4/5/24	PHC 780	male	1/1/83	NaN	Na
340361	e174bb5e-286f-4fa6-917b-4ff36d4ce130	4/5/24	937	female	1/1/69	NaN	Na
340362	3e365d82-1034-4a48-9dbe-1304888e3a8a	4/5/24	MAG 811	male	1/1/56	NaN	Na
340363	bd978cc4-778a-4ca6-90cf-d494dd3000d2	4/5/24	548	male	1/1/70	NaN	Na

340364 rows × 19 columns

```
# Display the first few rows
print(viral_load.head())

# Display the information about the DataFrame
print(viral_load.info())
```

```
2      12/16/82        8/26/10             9/21/10       10/25/11
3        1/1/63         8/9/11              8/9/11         8/9/11
4       4/12/82         9/4/08              9/4/08       10/14/10

  baseline_regimen  current_regimen  current_regimen_line  baseline_cd4  \
0      TDF-3TC-EFV              NaN                   NaN         384.0
1      D4T-3TC-NVP              NaN                   NaN           NaN
2      TDF-3TC-NVP              NaN                   NaN         292.0
3      AZT-3TC-NVP              NaN                   NaN         637.0
4      TDF-3TC-NVP              NaN                   NaN           NaN

  latest_cd4_date  latest_cd4 latest_viral_load_date  \
0          7/2/15       384.0                9/22/16
1        11/24/21      1151.0                1/30/24
2         6/17/15       750.0                    NaN
3          8/4/20      1181.0               10/17/23
4          9/3/13       500.0                 1/8/24

   latest_viral_load_copies latest_viral_load_qualitative  \
0                       NaN                           NaN
1                       NaN                           NaN
2                       NaN                           NaN
3                       1.0       BEYOND DETECTABLE LIMIT
```

```
Data columns (total 19 columns):
 #   Column                                Non-Null Count   Dtype  
---  ------                                --------------   -----  
 0   view_sentinel_events_id               340364 non-null  object 
 1   status_date                           340364 non-null  object 
 2   patient_clinic_no                     332399 non-null  object 
 3   gender                                340364 non-null  object 
 4   date_of_birth                         340364 non-null  object 
 5   diagnosis_date                        160971 non-null  object 
 6   art_enrollment_date                   161079 non-null  object 
 7   art_start_date                        173364 non-null  object 
 8   baseline_regimen                      168537 non-null  object 
 9   current_regimen                       0 non-null       float64
 10  current_regimen_line                  0 non-null       float64
 11  baseline_cd4                          60708 non-null   float64
 12  latest_cd4_date                       62367 non-null   object 
 13  latest_cd4                            62367 non-null   float64
 14  latest_viral_load_date                120539 non-null  object 
 15  latest_viral_load_copies              100030 non-null  float64
 16  latest_viral_load_qualitative         103896 non-null  object 
 17  latest_viral_load_suppression_status  106500 non-null  float64
 18  date_of_death                         13010 non-null   object 
dtypes: float64(6), object(13)
memory usage: 49.3+ MB
None
```

```
#VIRAL LOAD empty values

# Count NaN values
nan_count = viral_load['latest_viral_load_copies'].isna().sum()

# Count zero values
zero_count = (viral_load['latest_viral_load_copies'] == 0).sum()

# Count empty string values (if any)
empty_string_count = (viral_load['latest_viral_load_copies'] == '').sum()

# Total count of empty records
total_empty = nan_count + zero_count + empty_string_count

print(f"Number of NaN values: {nan_count}")
print(f"Number of zero values: {zero_count}")
print(f"Number of empty string values: {empty_string_count}")
print(f"Total number of empty records: {total_empty}")
print(f"Percentage of empty records: {(total_empty / len(viral_load)) * 100:.2f}%")
```

```
Number of NaN values: 240334
Number of zero values: 0
Number of empty string values: 0
Total number of empty records: 240334
Percentage of empty records: 70.61%
```

```
#CD4 Missing values

# Count NaN values
nan_count = viral_load['latest_cd4'].isna().sum()

# Count zero values
zero_count = (viral_load['latest_cd4'] == 0).sum()

# Count empty string values (if any)
empty_string_count = (viral_load['latest_cd4'] == '').sum()

# Total count of empty records
total_empty = nan_count + zero_count + empty_string_count

print(f"Number of NaN values: {nan_count}")
print(f"Number of zero values: {zero_count}")
print(f"Number of empty string values: {empty_string_count}")
print(f"Total number of empty records: {total_empty}")
print(f"Percentage of empty records: {(total_empty / len(viral_load)) * 100:.2f}%")
```

```
Number of NaN values: 277997
Number of zero values: 0
Number of empty string values: 0
Total number of empty records: 277997
Percentage of empty records: 81.68%
```

```
#checking for values where both cd4 and viral loads are empty or missing

def is_empty(value):
    """Check if a value is considered empty (NaN, 0, or empty string)"""
```

```
        if pd.isna(value):  # This checks for NaN
            return True
        if value == 0 or value == '':  # This checks for 0 and empty string
            return True
        return False

    # Count records where both latest_cd4 and latest_viral_load_copies are empty
    empty_count = viral_load.apply(lambda row: is_empty(row['latest_cd4']) and
                                               is_empty(row['latest_viral_load_copies']),
                                   axis=1).sum()

    total_records = len(viral_load)
    percentage_empty = (empty_count / total_records) * 100

    print(f"Number of records where both latest CD4 and viral load are empty: {empty_count}")
    print(f"Percentage of total records: {percentage_empty:.2f}%")
```

```
Number of records where both latest CD4 and viral load are empty: 212142
Percentage of total records: 62.33%
```

```
    #Count total number of records
    total_record_count = viral_load.shape[0]
    print(f"Number of records: {total_record_count}")
```

```
Number of records: 340364
```

```
    # Display data types and non-null counts
    print(viral_load.info())

    # Summary statistics
    print(viral_load.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 340364 entries, 0 to 340363
Data columns (total 19 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   view_sentinel_events_id            340364 non-null  object
 1   status_date                        340364 non-null  object
 2   patient_clinic_no                  332399 non-null  object
 3   gender                             340364 non-null  object
 4   date_of_birth                      340364 non-null  object
 5   diagnosis_date                     160971 non-null  object
 6   art_enrollment_date                161079 non-null  object
 7   art_start_date                     173364 non-null  object
 8   baseline_regimen                   168537 non-null  object
 9   current_regimen                    0 non-null       float64
 10  current_regimen_line               0 non-null       float64
 11  baseline_cd4                       60708 non-null   float64
 12  latest_cd4_date                    62367 non-null   object
 13  latest_cd4                         62367 non-null   float64
 14  latest_viral_load_date             120539 non-null  object
 15  latest_viral_load_copies           100030 non-null  float64
 16  latest_viral_load_qualitative      103896 non-null  object
 17  latest_viral_load_suppression_status  106500 non-null  float64
 18  date_of_death                      13010 non-null   object
dtypes: float64(6), object(13)
memory usage: 49.3+ MB
None
       current_regimen  current_regimen_line  baseline_cd4    latest_cd4  \
count              0.0                   0.0  60708.000000  62367.000000
mean               NaN                   NaN    344.308361    493.151057
std                NaN                   NaN    287.126052    327.280744
min                NaN                   NaN      0.000000      1.000000
25%                NaN                   NaN    175.000000    262.000000
50%                NaN                   NaN    281.000000    450.000000
75%                NaN                   NaN    447.000000    657.000000
max                NaN                   NaN  17930.000000  18527.000000

       latest_viral_load_copies  latest_viral_load_suppression_status
count              1.000300e+05                         106500.000000
mean               7.476265e+03                              0.939953
std                2.261195e+05                              0.237575
min                1.000000e+00                              0.000000
25%                1.000000e+00                              1.000000
50%                1.000000e+00                              1.000000
75%                1.100000e+02                              1.000000
max                4.210000e+07                              1.000000
```

```
    #fill the missing variables with values of mean

    # Replace missing values with the mean
    viral_load['latest_cd4'].fillna(viral_load['latest_cd4'].mean(), inplace=True)
    viral_load['latest_viral_load_copies'].fillna(viral_load['latest_viral_load_copies'].mean(), inplace=True)
```

```
# Replace missing values with the median
viral_load['latest_cd4'].fillna(viral_load['latest_cd4'].median(), inplace=True)
viral_load['latest_viral_load_copies'].fillna(viral_load['latest_viral_load_copies'].median(), inplace=True)

#PERFORM AG COUNT AGAIN AFTER FILLING EMPTY VALUES
# Again Count records where both latest_cd4 and latest_viral_load_copies are empty
empty_count = viral_load.apply(lambda row: is_empty(row['latest_cd4']) and
                                           is_empty(row['latest_viral_load_copies']),
                               axis=1).sum()

print(f"Number of records where both latest CD4 and viral load are empty: {empty_count}")
```

```
Number of records where both latest CD4 and viral load are empty: 0
```

```
#Adding more columns to my dataset
# Convert date columns to datetime
viral_load['date_of_birth'] = pd.to_datetime(viral_load['date_of_birth'], format='%m/%d/%y', errors='coerce')
viral_load['art_start_date'] = pd.to_datetime(viral_load['art_start_date'], format='%m/%d/%y', errors='coerce')

# Calculate age
viral_load['age'] = (pd.to_datetime('today') - viral_load['date_of_birth']).dt.days // 365

# Calculate treatment period
viral_load['treatment_period'] = (pd.to_datetime('today') - viral_load['art_start_date']).dt.days // 365

# Add a column for viral load status
viral_load['viral_load_status'] = viral_load['latest_viral_load_copies'].apply(lambda x: 'Suppressed' if x < 1000 else 'Not
```

```
# Display all columns in the DataFrame
print(viral_load.columns.tolist())
```

```
['view_sentinel_events_id', 'status_date', 'patient_clinic_no', 'gender', 'date_of_birth', 'diagnosis_date', 'art_enroll
```

```
#removing unwanted columns
# Remove the 'view_sentinel_events_id' column
viral_load.drop(columns=['view_sentinel_events_id'], inplace=True)
```

```
# Again after removing
print(viral_load.columns.tolist())
```

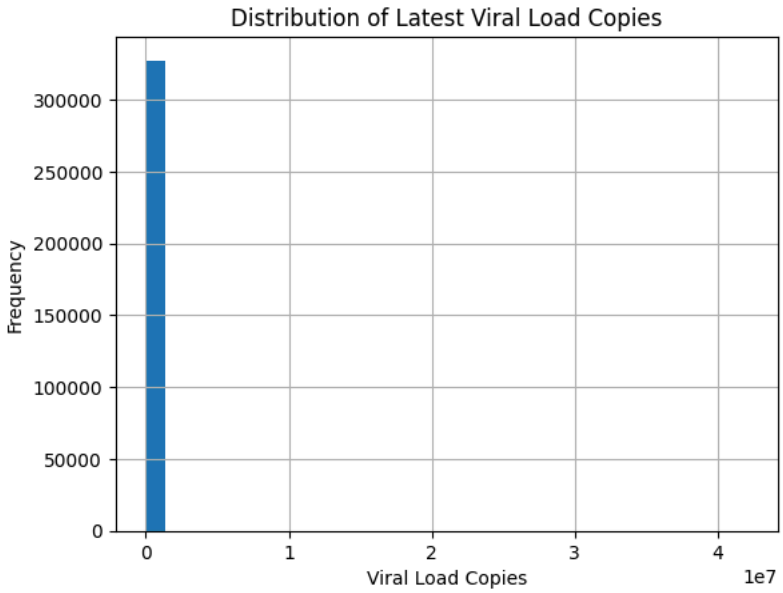
```
['status_date', 'patient_clinic_no', 'gender', 'date_of_birth', 'diagnosis_date', 'art_enrollment_date', 'art_start_date
```

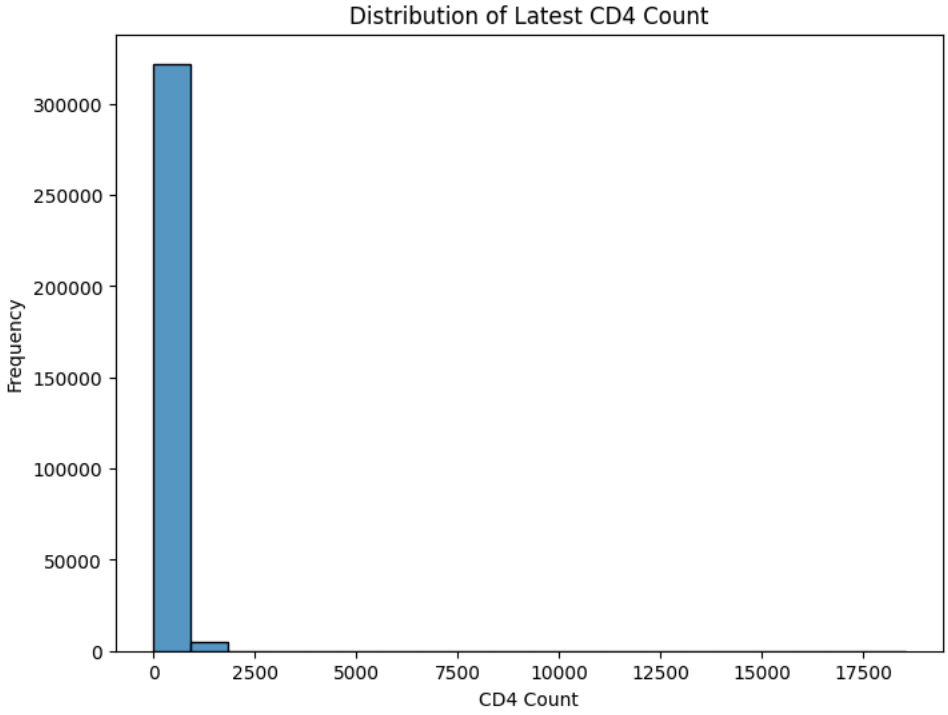
```
# Remove rows where 'date_of_death' is not null
viral_load_cleaned = viral_load[viral_load['date_of_death'].isnull()]

# Display the shape of the cleaned DataFrame
print(viral_load_cleaned.shape)

# Display the first few rows of the cleaned DataFrame
print(viral_load_cleaned.head())
```

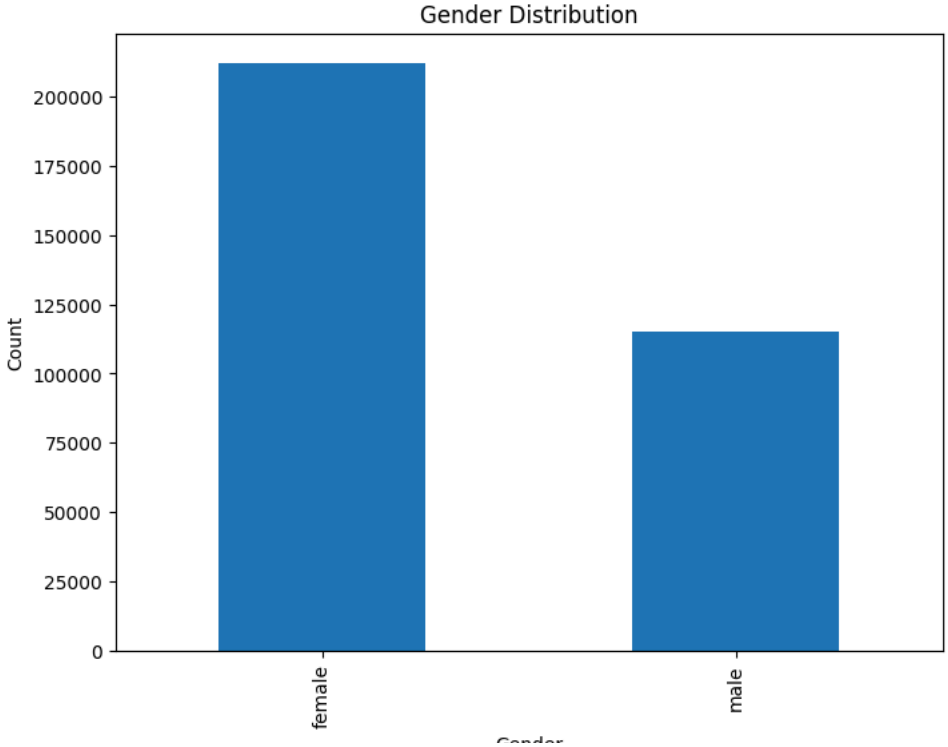
```
(327354, 21)
  status_date patient_clinic_no  gender date_of_birth diagnosis_date  \
0      4/5/24             16253  female    1994-01-01            NaN
1      4/5/24               621  female    2008-12-01        8/12/09
2      4/5/24          DEMO1940  female    1982-12-16        8/26/10
3      4/5/24                37  female    2063-01-01         8/9/11
4      4/5/24           ACH0700  female    1982-04-12         9/4/08

  art_enrollment_date art_start_date baseline_regimen  current_regimen  \
0                 NaN     2015-07-02      TDF-3TC-EFV              NaN
1             8/12/09     2010-03-10      D4T-3TC-NVP              NaN
2             9/21/10     2011-10-25      TDF-3TC-NVP              NaN
3              8/9/11     2011-08-09      AZT-3TC-NVP              NaN
4              9/4/08     2010-10-14      TDF-3TC-NVP              NaN

   current_regimen_line  ...  latest_cd4_date latest_cd4  \
0                   NaN  ...           7/2/15      384.0
1                   NaN  ...         11/24/21     1151.0
2                   NaN  ...          6/17/15      750.0
3                   NaN  ...           8/4/20     1181.0
4                   NaN  ...           9/3/13      500.0
```

```
   latest_viral_load_date latest_viral_load_copies  \
0                9/22/16              7476.26543
1                1/30/24              7476.26543
2                    NaN              7476.26543
3               10/17/23                 1.00000
4                 1/8/24                79.00000

   latest_viral_load_qualitative latest_viral_load_suppression_status  \
0                            NaN                                   NaN
1                            NaN                                   NaN
2                            NaN                                   NaN
3       BEYOND DETECTABLE LIMIT                                   1.0
4       BEYOND DETECTABLE LIMIT                                   1.0

   date_of_death age  treatment_period  viral_load_status
0            NaN  30               9.0     Not Suppressed
1            NaN  15              14.0     Not Suppressed
2            NaN  41              12.0     Not Suppressed
3            NaN -39              13.0         Suppressed
4            NaN  42              13.0         Suppressed

[5 rows x 21 columns]
```

```
#THIS IS THE END OF DATA CLEANING PROCESS , NOW LET'S GAIN INSIGHT INTO THE DATA BY USING SOME VISUALS


# Histogram for numerical variables

import matplotlib.pyplot as plt
import seaborn as sns

# Histogram for numerical variablees (latest_viral_load_copies)
viral_load_cleaned['latest_viral_load_copies'].hist(bins=30)
plt.title('Distribution of Latest Viral Load Copies')
plt.xlabel('Viral Load Copies')
plt.ylabel('Frequency')
plt.show()

# Histogram for numerical variables(latest_cd4)
plt.figure(figsize=(8, 6))
sns.histplot(data=viral_load_cleaned, x="latest_cd4", bins=20)
plt.title("Distribution of Latest CD4 Count")
plt.xlabel("CD4 Count")
plt.ylabel("Frequency")
plt.show()

# Bar plot for categorical variables
plt.figure(figsize=(8, 6))
viral_load_cleaned['gender'].value_counts().plot(kind='bar')
plt.title("Gender Distribution")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.show()
```


Distribution of Latest Viral Load Copies
Frequency
Viral Load Copies
1e7


Distribution of Latest CD4 Count
Frequency
CD4 Count


Gender Distribution
Count
female
male

Gender

```
# Scatter plot for numerical variables
plt.figure(figsize=(8, 6))
plt.scatter(viral_load_cleaned['latest_cd4'], viral_load_cleaned['latest_viral_load_copies'])
plt.title("Latest CD4 vs Latest Viral Load")
plt.xlabel("Latest CD4 Count")
plt.ylabel("Latest Viral Load Copies")
plt.show()

# Box plot for categorical vs numerical
plt.figure(figsize=(8, 6))
sns.boxplot(x="gender", y="latest_cd4", data=viral_load_cleaned)
plt.title("Latest CD4 Count by Gender")
plt.xlabel("Gender")
plt.ylabel("Latest CD4 Count")
plt.show()
```