```
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
file_path = "/content/drive/MyDrive/Projects/Exercise Files/Demo Hospital Outpatient Data_NHC.csv"
```

```
df_visit = pd.read_csv(file_path)
```

```
df_visit.head()
```

	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Total_
0	2020-05-06	688923	68	Female	Diabetes	True	20006	22
1	2018-08-04	886361	62	Female	Urinary Tract Infection	False	20005	34
2	2021-04-10	464823	70	Female	Upper Respiratory Tract Infection	True	10003	18
					Upper			

```
df_visit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 13 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   Visit_Date     1000000 non-null  object
 1   Patient_ID     1000000 non-null  int64
 2   Age            1000000 non-null  int64
 3   Gender         1000000 non-null  object
 4   Diagnosis      1000000 non-null  object
 5   Has_Insurance  1000000 non-null  bool
 6   Postcode       1000000 non-null  int64
 7   Total_Cost     1000000 non-null  float64
 8   Registration   1000000 non-null  int64
 9   Nursing        1000000 non-null  int64
 10  Laboratory     1000000 non-null  int64
 11  Consultation   1000000 non-null  int64
 12  Pharmacy       1000000 non-null  int64
dtypes: bool(1), float64(1), int64(8), object(3)
memory usage: 92.5+ MB
```

```
df_visit.isna().sum()
```

```
Visit_Date       0
Patient_ID       0
Age              0
Gender           0
Diagnosis        0
Has_Insurance    0
Postcode         0
Total_Cost       0
Registration     0
Nursing          0
Laboratory       0
Consultation     0
Pharmacy         0
dtype: int64
```

```
df_visit.rename(columns= {'Registration': 'Registration_minutes', 'Nursing' : 'Nursing_minutes','Laboratory': 'Laboratory_minute
```

```
df_visit.columns
```

```
Index(['Visit_Date', 'Patient_ID', 'Age', 'Gender', 'Diagnosis',
       'Has_Insurance', 'Postcode', 'Total_Cost', 'Registration_minutes',
       'Nursing_minutes', 'Laboratory_minutes', 'Consultation_minutes',
       'Pharmacy_minutes'],
      dtype='object')
```

```
df_visit['Visit_Date'] = pd.to_datetime(df_visit['Visit_Date'])
```

```
df_visit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 13 columns):
 #   Column                Non-Null Count    Dtype
---  ------                --------------    -----
 0   Visit_Date            1000000 non-null  datetime64[ns]
 1   Patient_ID            1000000 non-null  int64
 2   Age                   1000000 non-null  int64
 3   Gender                1000000 non-null  object
 4   Diagnosis             1000000 non-null  object
 5   Has_Insurance         1000000 non-null  bool
 6   Postcode              1000000 non-null  int64
 7   Total_Cost            1000000 non-null  float64
 8   Registration_minutes  1000000 non-null  int64
 9   Nursing_minutes       1000000 non-null  int64
 10  Laboratory_minutes    1000000 non-null  int64
 11  Consultation_minutes  1000000 non-null  int64
 12  Pharmacy_minutes      1000000 non-null  int64
dtypes: bool(1), datetime64[ns](1), float64(1), int64(8), object(2)
memory usage: 92.5+ MB
```

```
df_visit.head()
```

	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Total_
0	2020-05-06	688923	68	Female	Diabetes	True	20006	22
1	2018-08-04	886361	62	Female	Urinary Tract Infection	False	20005	34
2	2021-04-10	464823	70	Female	Upper Respiratory Tract Infection	True	10003	18
3	2021-10-01	655214	8	Female	Upper Respiratory Tract Infection	False	10006	32
4	2018-04-30	454666	24	Male	Malaria	True	10006	22

```
# Saving data_frame into different format
```

```
df_visit.to_csv('/content/drive/MyDrive/Projects/Exercise Files/Demo Hospital Outpatient Data_NHC_transformed.csv',index=False)
```

```
import zipfile
```

```
zipped_file_name = 'demo_hospital_outpatient_data_transformed.zip'
```

```
with zipfile.ZipFile(zipped_file_name,'w',zipfile.ZIP_DEFLATED) as zip:
    zip.write('/content/drive/MyDrive/Projects/Exercise Files/Demo Hospital Outpatient Data_NHC_transformed.csv')
```

```
pd.read_csv('demo_hospital_outpatient_data_transformed.zip')
```

	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Tc
0	2020-05-06	688923	68	Female	Diabetes	True	20006	
1	2018-08-04	886361	62	Female	Urinary Tract Infection	False	20005	
2	2021-04-10	464823	70	Female	Upper Respiratory Tract Infection	True	10003	
3	2021-10-01	655214	8	Female	Upper Respiratory Tract Infection	False	10006	
4	2018-04-30	454666	24	Male	Malaria	True	10006	
...	...	...	...	...	...	...	...	
999995	2018-09-30	385435	9	Male	Abdominal pain	True	10004	
999996	2020-05-08	117261	29	Female	Urinary Tract Infection	True	20009	
999997	2019-12-31	594613	39	Female	Upper Respiratory Tract Infection	False	10001	
999998	2019-11-04	152179	39	Female	Malaria	False	20006	
999999	2019-05-11	370584	76	Female	Malaria	False	20012	

1000000 rows × 13 columns

```
!pip install pyreadstat
import pyreadstat
```

```
Collecting pyreadstat
  Downloading pyreadstat-1.2.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 15.3 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from pyreadstat) (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2.0->pyread
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2.0->pyreadstat) (202
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2.0->pyreadstat) (2
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2.0->pyreadstat) (1.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.2
Installing collected packages: pyreadstat
Successfully installed pyreadstat-1.2.7
```

```
spss_file = '/content/drive/MyDrive/Projects/Exercise Files/Demo Hospital Outpatient Data_NHC_transformed.sav'
```

```
pyreadstat.write_sav(df_visit,spss_file)
```

```
data, meta = pyreadstat.read_sav('/content/drive/MyDrive/Projects/Exercise Files/Demo Hospital Outpatient Data_NHC_transformed.s
```

```
data
```

	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	T
0	2020-05-06	688923.0	68.0	Female	Diabetes	1.0	20006.0	
1	2018-08-04	886361.0	62.0	Female	Urinary Tract Infection	0.0	20005.0	
2	2021-04-10	464823.0	70.0	Female	Upper Respiratory Tract Infection	1.0	10003.0	
3	2021-10-01	655214.0	8.0	Female	Upper Respiratory Tract Infection	0.0	10006.0	
4	2018-04-30	454666.0	24.0	Male	Malaria	1.0	10006.0	
...	...	...	...	...	...	...	...	
999995	2018-09-30	385435.0	9.0	Male	Abdominal pain	1.0	10004.0	
999996	2020-05-08	117261.0	29.0	Female	Urinary Tract Infection	1.0	20009.0	
999997	2019-12-31	594613.0	39.0	Female	Upper Respiratory Tract Infection	0.0	10001.0	
999998	2019-11-04	152179.0	39.0	Female	Malaria	0.0	20006.0	
999999	2019-05-11	370584.0	76.0	Female	Malaria	0.0	20012.0	

1000000 rows × 13 columns

```
#EDA of Hospital Outcome
```

```
import matplotlib.pyplot as plt
```

```
df_visit['Gender'].value_counts()
```

```
Gender
Male      500108
Female    499892
Name: count, dtype: int64
```

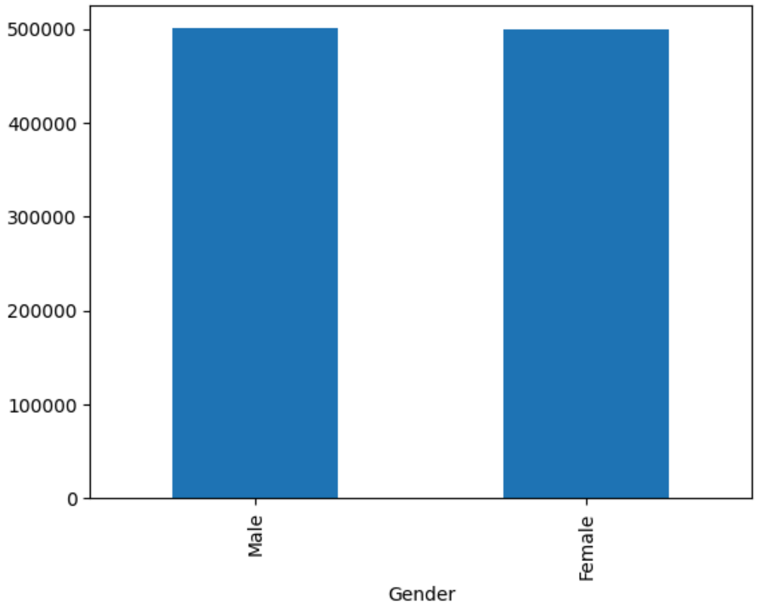
```
visit_gender_count = df_visit['Gender'].value_counts()
```

```
visit_gender_count
```

```
Gender
Male      500108
Female    499892
Name: count, dtype: int64
```

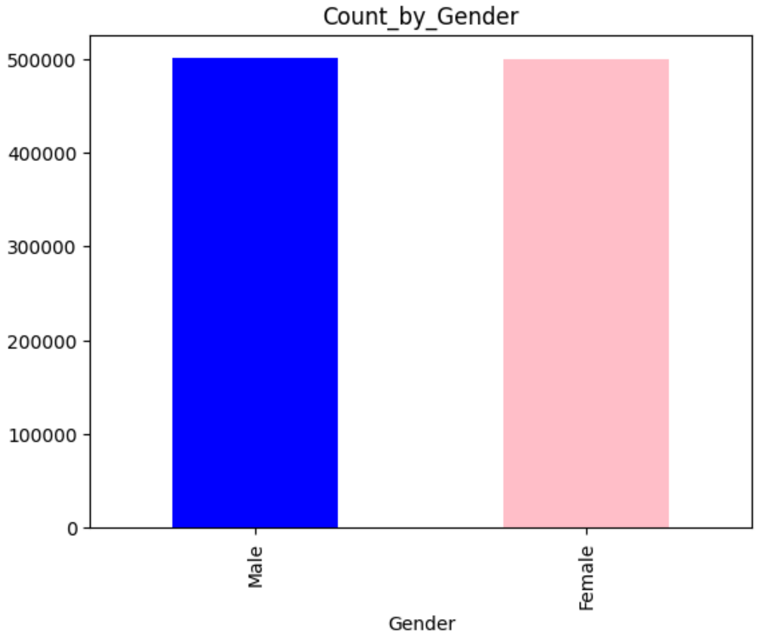
```
visit_gender_count.plot(kind='bar')
```

```
<Axes: xlabel='Gender'>
```

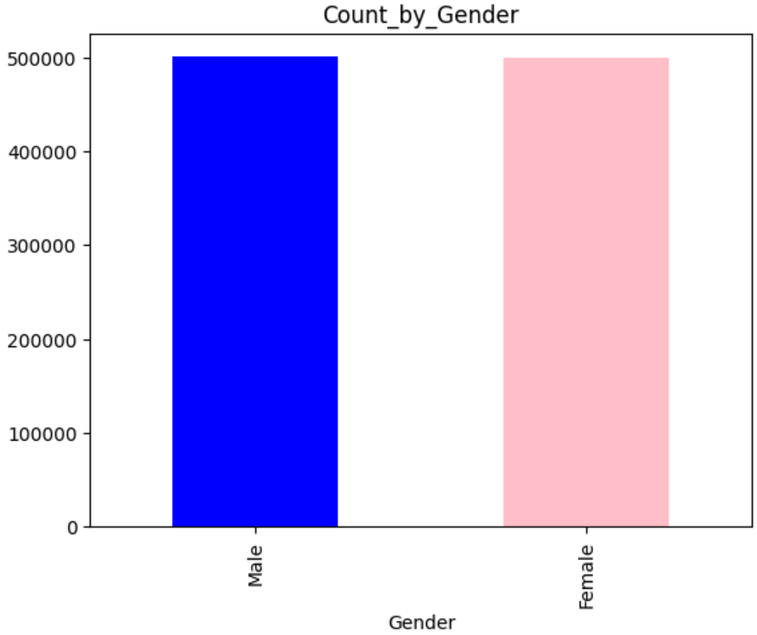


```
visit_gender_count.plot(kind='bar', color =['Blue','Pink'])
plt.title("Count_by_Gender")
```

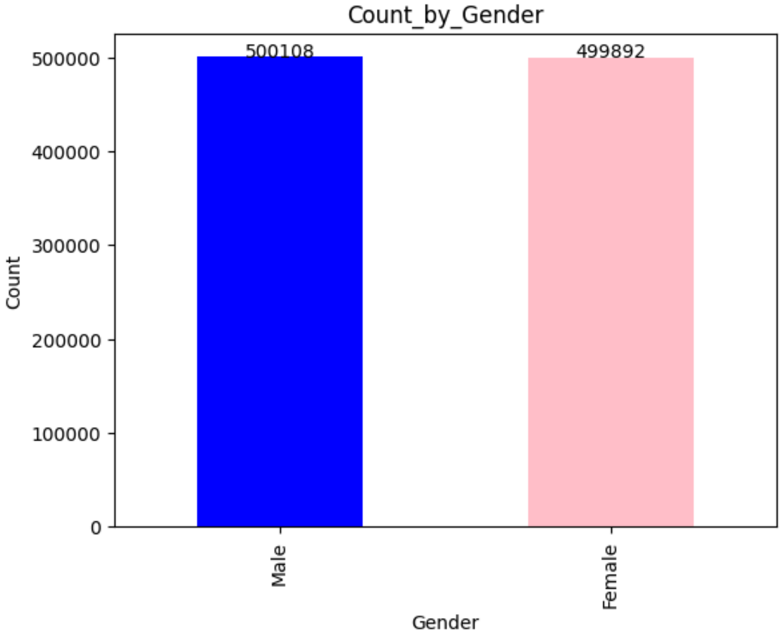
```
Text(0.5, 1.0, 'Count_by_Gender')
```



```
visit_gender_count.plot(kind='bar', color =['Blue','Pink'])
plt.title("Count_by_Gender")
plt.show()
```



```
chart = visit_gender_count.plot(kind='bar', color =['Blue','Pink'])
plt.title("Count_by_Gender")
plt.ylabel("Count")
for i, count in enumerate(visit_gender_count):
    chart.text(i, count + 0.1, str(count), ha='center')
```



```
df_visit['Age'].max()
```

```
90
```

```
df_visit['Age'].min()
```

```
0
```

```
Age_range = [0,10,20,30,40,50,60,70,80,90]
```

```
df_visit['Agerange']= pd.cut(df_visit['Age'],bins = Age_range)
```

```
df_visit.head()
```

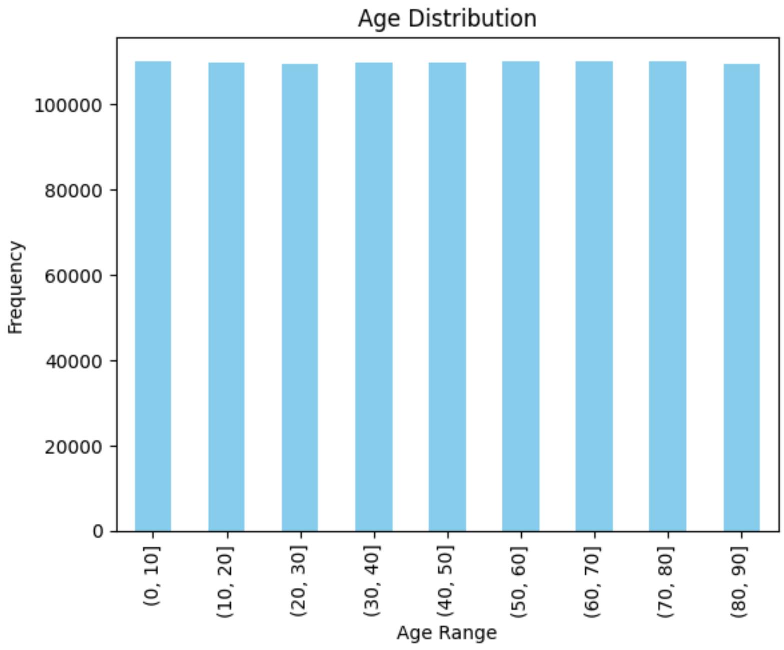
	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Total_
0	2020-05-06	688923	68	Female	Diabetes	True	20006	22
1	2018-08-04	886361	62	Female	Urinary Tract Infection	False	20005	34
2	2021-04-10	464823	70	Female	Upper Respiratory Tract Infection	True	10003	18
3	2021-10-01	655214	8	Female	Upper Respiratory Tract Infection	False	10006	32
4	2018-04-30	454666	24	Male	Malaria	True	10006	22

```
df_visit['Agerange'].value_counts().sort_index()
```

```
Agerange
(0, 10]     110025
(10, 20]    109930
(20, 30]    109577
(30, 40]    109723
(40, 50]    109960
(50, 60]    110259
(60, 70]    110039
(70, 80]    110051
(80, 90]    109515
Name: count, dtype: int64
```

```
age_distribution = df_visit['Agerange'].value_counts().sort_index()
```

```
age_distribution.plot(kind = 'bar',color = 'Skyblue')
plt.title("Age Distribution")
plt.xlabel('Age Range')
plt.ylabel('Frequency')
plt.show()
```



```
postcode_counts = df_visit['Postcode'].value_counts()
```

```
postcode_counts
```

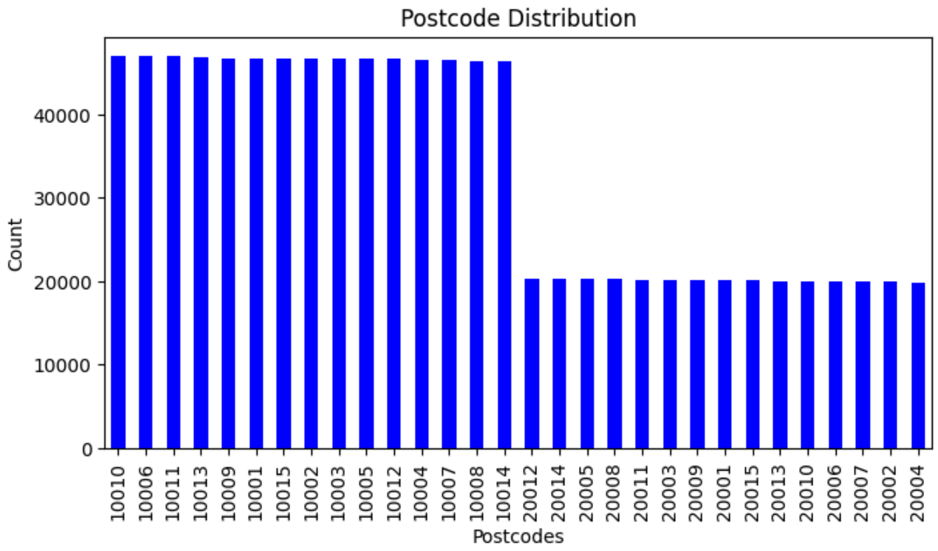
```
Postcode
10010    46924
```

```
10006    46919
10011    46896
10013    46735
10009    46700
10001    46696
10015    46666
10002    46608
10003    46600
10005    46584
10012    46569
10004    46486
10007    46452
10008    46298
10014    46252
20012    20322
20014    20244
20005    20205
20008    20196
20011    20118
20003    20096
20009    20042
20001    20031
20015    20030
20013    19954
20010    19950
20006    19895
20007    19879
20002    19858
20004    19795
Name: count, dtype: int64
```

```
df_visit['Postcode'].nunique()
```

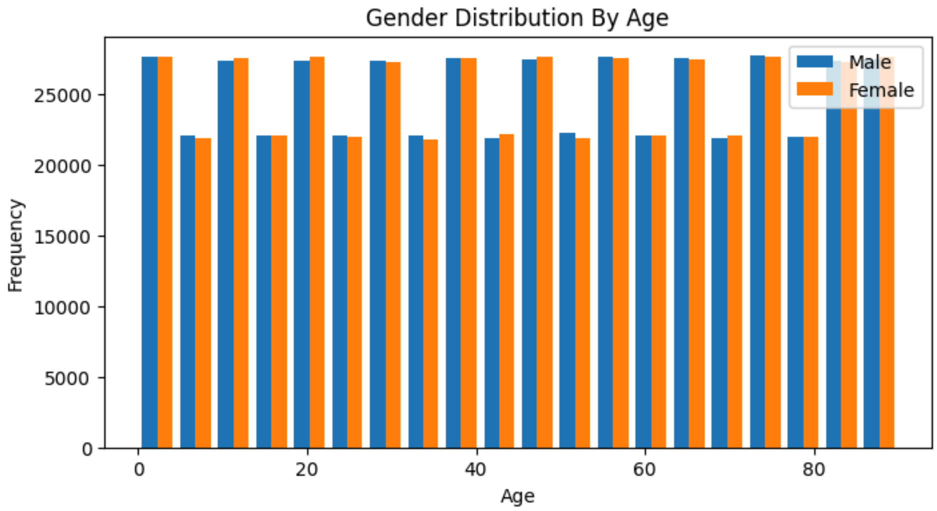
```
30
```

```
plt.figure(figsize= (8,4))
postcode_counts.plot(kind='bar', color='Blue')
plt.title("Postcode Distribution")
plt.xlabel("Postcodes")
plt.ylabel("Count")
plt.show()
```



```
# BIVARIATE ANALYSIS
```

```
plt.figure(figsize = (8,4))
plt.hist([df_visit[df_visit['Gender']=='Male']['Age'],df_visit[df_visit['Gender']=='Female']['Age']],bins= 20, label = ['Male','F
plt.title("Gender Distribution By Age")
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```



```
# Trend Analysis
```

```
print(df_visit['Visit_Date'].dtype)
```

```
datetime64[ns]
```

```
df_visit.head()
```

	Visit_Date	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Total_
0	2020-05-06	688923	68	Female	Diabetes	True	20006	2
1	2018-08-04	886361	62	Female	Urinary Tract Infection	False	20005	3
2	2021-04-10	464823	70	Female	Upper Respiratory Tract Infection	True	10003	1
3	2021-10-01	655214	8	Female	Upper Respiratory Tract Infection	False	10006	3
4	2018-04-30	454666	24	Male	Malaria	True	10006	2

```
df_visit['Visit_Date'].nunique()
```

```
1825
```

```
df_visit['Visit_Date'].value_counts().sort_values(ascending=False)
```

```
Visit_Date
2022-04-05    630
2018-10-28    623
2020-10-22    621
2020-07-22    614
2020-06-03    612
             ...
2021-03-11    483
2019-06-08    483
2019-04-10    482
2018-11-18    481
2022-11-11    469
Name: count, Length: 1825, dtype: int64
```

```
df_visit['Visit_Date'].max()
```

```
Timestamp('2022-12-30 00:00:00')
```

```
df_visit['Visit_Date'].min()
```

```
Timestamp('2018-01-01 00:00:00')
```

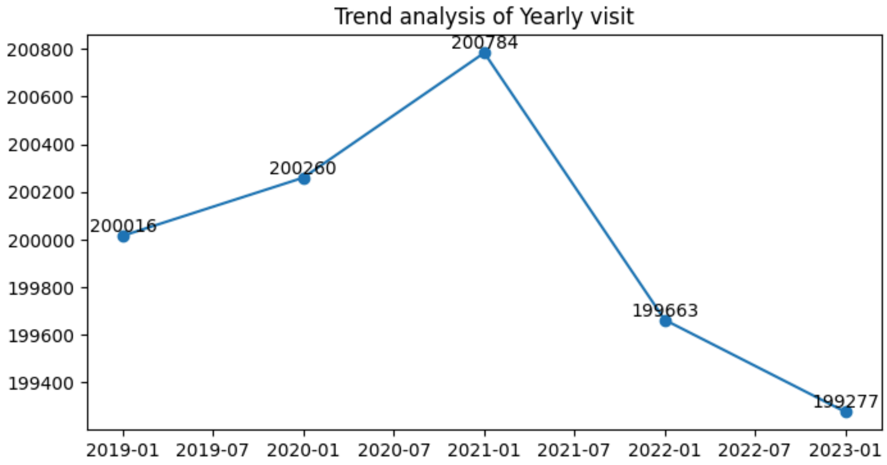
```
df_visit.set_index('Visit_Date',inplace=True)
```

```
df_resampled_year = df_visit.resample("Y").count()
```

```
df_resampled_year
```

	Patient_ID	Age	Gender	Diagnosis	Has_Insurance	Postcode	Total_
Visit_Date							
2018-12-31	200016	200016	200016	200016	200016	200016	20
2019-12-31	200260	200260	200260	200260	200260	200260	20
2020-12-31	200784	200784	200784	200784	200784	200784	20
2021-12-31	199663	199663	199663	199663	199663	199663	19
2022-12-31	199277	199277	199277	199277	199277	199277	19

```
plt.figure(figsize=(8,4))
plt.plot(df_resampled_year.index,df_resampled_year['Patient_ID'],marker='o', linestyle='-')
plt.title("Trend analysis of Yearly visit")
for i, count in enumerate(df_resampled_year['Patient_ID']):
    plt.text(df_resampled_year.index[i], count, str(count), ha='center', va='bottom')
plt.show()
```



```
df_resample_month = df_visit.resample("M").count()
```

```
df_resample_test = df_visit.resample("M")
```

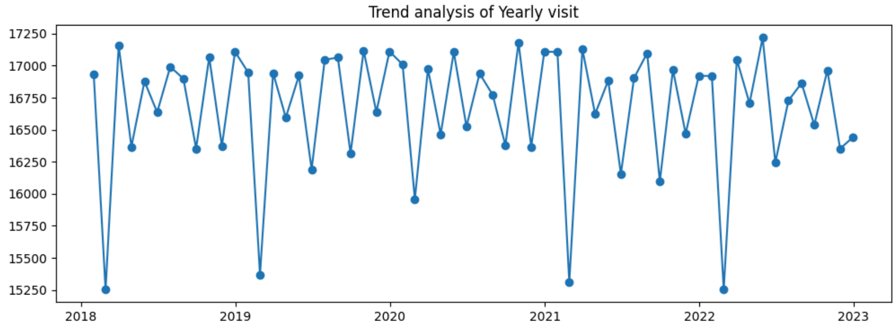
```
print(df_resample_test)
```

```
DatetimeIndexResampler [freq=<MonthEnd>, axis=0, closed=right, label=right, convention=start, origin=start_day]
```

```
df_resample_month
```

```
2020-05-31   17107  17107  17107  17107  17107  17107  17
2020-06-30   16525  16525  16525  16525  16525  16525  16
2020-07-31   16939  16939  16939  16939  16939  16939  16
2020-08-31   16772  16772  16772  16772  16772  16772  16
2020-09-30   16377  16377  16377  16377  16377  16377  16
2020-10-31   17180  17180  17180  17180  17180  17180  17
2020-11-30   16367  16367  16367  16367  16367  16367  16
2020-12-31   17110  17110  17110  17110  17110  17110  17
2021-01-31   17107  17107  17107  17107  17107  17107  17
2021-02-28   15307  15307  15307  15307  15307  15307  15
2021-03-31   17130  17130  17130  17130  17130  17130  17
2021-04-30   16623  16623  16623  16623  16623  16623  16
2021-05-31   16883  16883  16883  16883  16883  16883  16
2021-06-30   16155  16155  16155  16155  16155  16155  16
2021-07-31   16903  16903  16903  16903  16903  16903  16
2021-08-31   17098  17098  17098  17098  17098  17098  17
2021-09-30   16098  16098  16098  16098  16098  16098  16
2021-10-31   16969  16969  16969  16969  16969  16969  16
2021-11-30   16469  16469  16469  16469  16469  16469  16
2021-12-31   16921  16921  16921  16921  16921  16921  16
2022-01-31   16920  16920  16920  16920  16920  16920  16
2022-02-28   15252  15252  15252  15252  15252  15252  15
2022-03-31   17048  17048  17048  17048  17048  17048  17
2022-04-30   16706  16706  16706  16706  16706  16706  16
2022-05-31   17219  17219  17219  17219  17219  17219  17
2022-06-30   16244  16244  16244  16244  16244  16244  16
2022-07-31   16728  16728  16728  16728  16728  16728  16
2022-08-31   16864  16864  16864  16864  16864  16864  16
2022-09-30   16537  16537  16537  16537  16537  16537  16
2022-10-31   16964  16964  16964  16964  16964  16964  16
2022-11-30   16352  16352  16352  16352  16352  16352  16
2022-12-31   16443  16443  16443  16443  16443  16443  16
```

```
plt.figure(figsize=(12,4))
plt.plot(df_resample_month.index,df_resample_month['Patient_ID'],marker='o', linestyle='-')
plt.title("Trend analysis of Yearly visit")

plt.show()
```



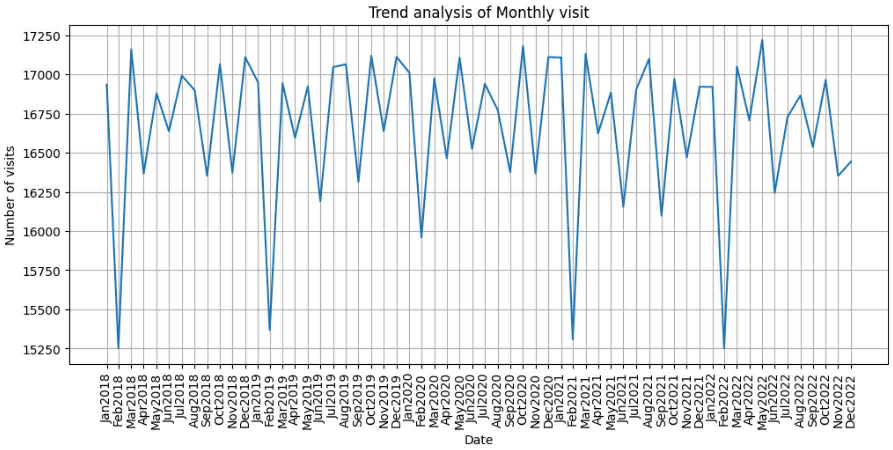
```
#Monthly Trend Analysis
```

```
monthly_ticks = pd.date_range(start= df_resample_month.index.min(),end = df_resample_month.index.max(),freq="M")
```

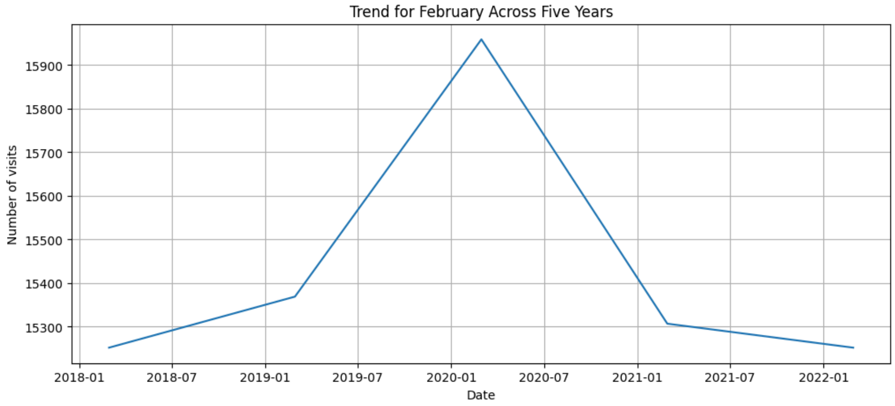
```
monthly_ticks
```

```
    DatetimeIndex(['2018-01-31', '2018-02-28', '2018-03-31', '2018-04-30',
                   '2018-05-31', '2018-06-30', '2018-07-31', '2018-08-31',
                   '2018-09-30', '2018-10-31', '2018-11-30', '2018-12-31',
                   '2019-01-31', '2019-02-28', '2019-03-31', '2019-04-30',
                   '2019-05-31', '2019-06-30', '2019-07-31', '2019-08-31',
                   '2019-09-30', '2019-10-31', '2019-11-30', '2019-12-31',
                   '2020-01-31', '2020-02-29', '2020-03-31', '2020-04-30',
                   '2020-05-31', '2020-06-30', '2020-07-31', '2020-08-31',
                   '2020-09-30', '2020-10-31', '2020-11-30', '2020-12-31',
                   '2021-01-31', '2021-02-28', '2021-03-31', '2021-04-30',
                   '2021-05-31', '2021-06-30', '2021-07-31', '2021-08-31',
                   '2021-09-30', '2021-10-31', '2021-11-30', '2021-12-31',
                   '2022-01-31', '2022-02-28', '2022-03-31', '2022-04-30',
                   '2022-05-31', '2022-06-30', '2022-07-31', '2022-08-31',
                   '2022-09-30', '2022-10-31', '2022-11-30', '2022-12-31'],
                  dtype='datetime64[ns]', freq='M')
```

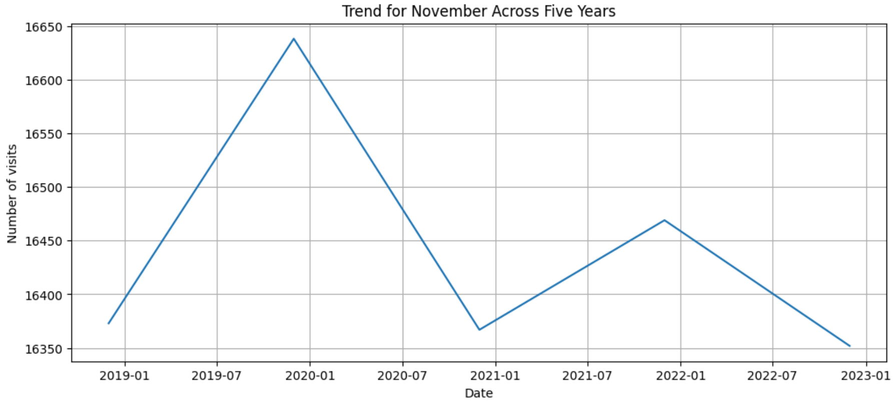
```
monthly_labels = [date.strftime('%b%Y') for date in monthly_ticks]



plt.figure(figsize=(12,5))
plt.plot(df_resample_month.index, df_resample_month['Patient_ID'])
plt.title("Trend analysis of Monthly visit")
plt.xticks(monthly_ticks,monthly_labels,rotation=90)
plt.xlabel('Date')
plt.ylabel('Number of visits')
plt.grid(True)
plt.show()
```



```
target_month = 2

df_target_month = df_resample_month[df_resample_month.index.month == target_month]
plt.figure(figsize=(12,5))
plt.plot(df_target_month.index, df_target_month['Postcode'])
plt.title(f'Trend for {df_target_month.index[0].strftime("%B")} Across Five Years')
plt.xlabel('Date')
plt.ylabel('Number of visits')
plt.grid(True)
```



```
target_month = 11

df_target_month = df_resample_month[df_resample_month.index.month == target_month]
plt.figure(figsize=(12,5))
plt.plot(df_target_month.index, df_target_month['Postcode'])
plt.title(f'Trend for {df_target_month.index[0].strftime("%B")} Across Five Years')
plt.xlabel('Date')
plt.ylabel('Number of visits')
plt.grid(True)
```



```
#Wait Time Analysis

df_visit['Total_minutes_minutes']= df_visit['Consultation_minutes'] + df_visit['Laboratory_minutes'] + df_visit['Nursing_minutes

df_visit['Total_Time'] = (df_visit['Total_minutes_minutes']/60).round(0)

df_visit.head()
```