

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving Appoitment.png to Appoitment.png

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving MED_APPOINTMENT.csv to MED_APPOINTMENT.csv

```
import os
print(os.listdir('.'))
```

['.config', 'Appoitment.png', 'MED_APPOINTMENT.csv', 'sample_data']

```
from IPython.display import Image, display
display(Image('Appoitment.png'))
```



```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv('MED_APPOINTMENT.csv')
```

```
df.head()
```

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	5.589980e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	4.262960e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	8.679510e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	8.841190e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1	0	0	0	No

```
df.dtypes
```

	0
PatientId	float64
AppointmentID	int64
Gender	object
ScheduledDay	object
AppointmentDay	object
Age	int64
Neighbourhood	object
Scholarship	int64
Hipertension	int64
Diabetes	int64
Alcoholism	int64
Handcap	int64
SMS_received	int64
No-show	object

dtype: object

```
df.columns
```

```
Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
      'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hipertension',
      'Diabetes', 'Alcoholism', 'Handcap', 'SMS_received', 'No-show'],
      dtype='object')
```

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

```

      0
PatientId    0
AppointmentID 0
Gender       0
ScheduledDay 0
AppointmentDay 0
Age          0
Neighbourhood 0
Scholarship  0
Hipertension 0
Diabetes     0
Alcoholism   0
Handcap      0
SMS_received 0
No-show      0

```

dtype: int64

```

#As displayed above there is no missing values but including the command for simulation
missing_percentage = df.isnull().mean() * 100
threshold = 30
columns_to_drop = missing_percentage[missing_percentage > threshold].index

```

```
df = df.drop(columns=columns_to_drop)
```

```

print(f"Columns dropped: {list(columns_to_drop)}")
print(f"Remaining columns: {df.columns}")

```

```

Columns dropped: []
Remaining columns: Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
                          'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hipertension',
                          'Diabetes', 'Alcoholism', 'Handcap', 'SMS_received', 'No-show'],
                          dtype='object')

```

```

from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in df.select_dtypes(include=['object']).columns:
    df[column] = label_encoder.fit_transform(df[column])
df.head()

```

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	2.987250e+13	5642903	0	27741	0	62	39	0	1	0	0	0	0	0
1	5.589980e+14	5642503	1	27503	0	56	39	0	0	0	0	0	0	0
2	4.262960e+12	5642549	0	27538	0	62	45	0	0	0	0	0	0	0
3	8.679510e+11	5642828	0	27708	0	8	54	0	0	0	0	0	0	0
4	8.841190e+12	5642494	0	27497	0	56	39	0	1	1	0	0	0	0

```

for column in df.select_dtypes(include=['float64']).columns:
    mean_value = df[column].mean()
    df[column] = df[column].fillna(mean_value)

for column in df.select_dtypes(include=['object']).columns:
    mode_value = df[column].mode()[0]
    df[column] = df[column].fillna(mode_value)

```

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

```

      0
PatientId    0
AppointmentID 0
Gender       0
ScheduledDay 0
AppointmentDay 0
Age          0
Neighbourhood 0
Scholarship  0
Hipertension 0
Diabetes     0
Alcoholism   0
Handcap      0
SMS_received 0
No-show      0

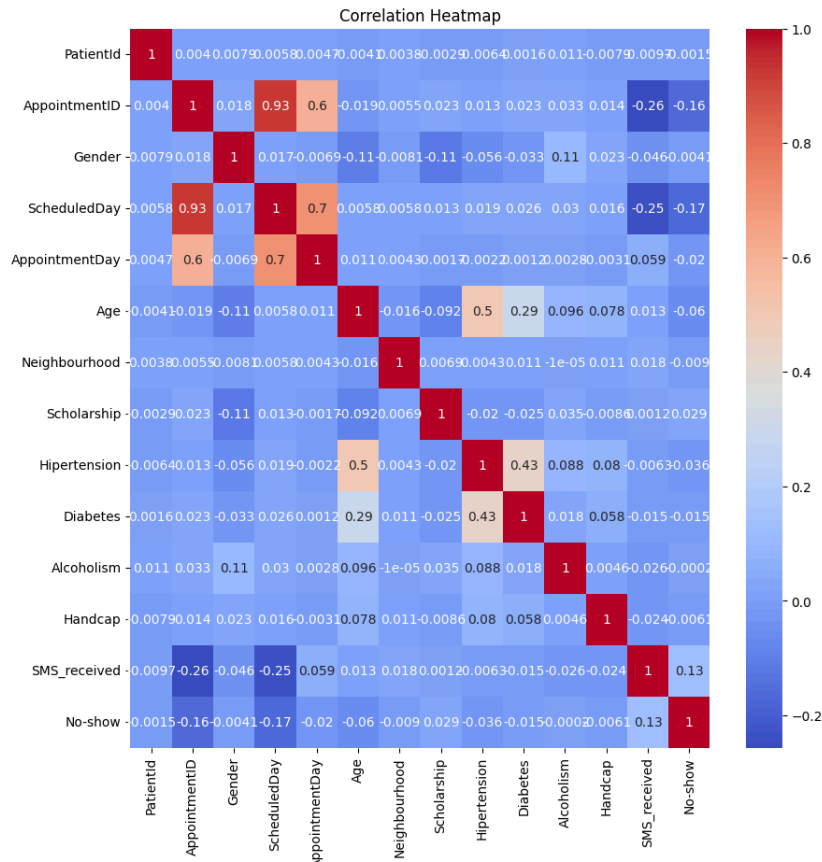
```

dtype: int64

```

plt.figure(figsize=(10, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

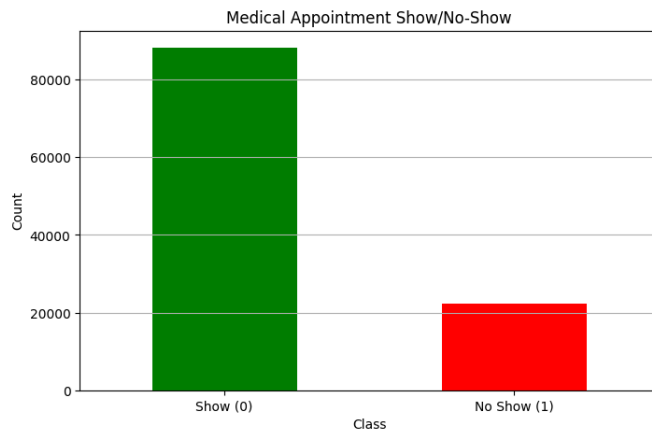
```



```
df['No-show'].value_counts()
```

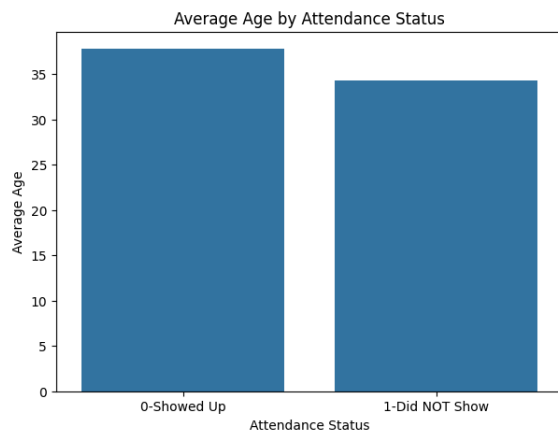
```
count
No-show
0      88208
1      22319
dtype: int64
```

```
plt.figure(figsize=(8, 5))
df['No-show'].value_counts().plot(kind='bar', color=['Green', 'Red'])
plt.xticks([0, 1], ['Show (0)', 'No Show (1)'], rotation=0)
plt.title('Medical Appointment Show/No-Show')
plt.xlabel('Class')
plt.ylabel('Count')
plt.grid(axis='y')
plt.show()
```

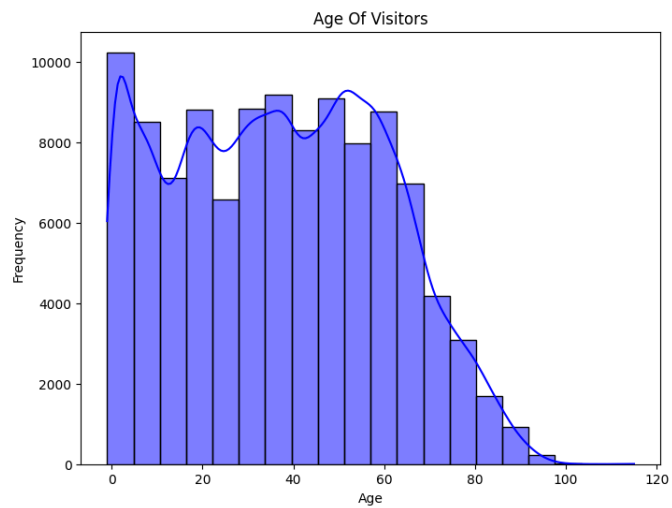


```
plt.figure(figsize=(7,5))
sns.barplot(x='No-show', y='Age', data=df, estimator='mean', ci=None)
plt.title("Average Age by Attendance Status")
plt.xlabel("Attendance Status")
plt.ylabel("Average Age")
plt.xticks([0, 1], ['0-Showed Up', '1-Did NOT Show'])
```

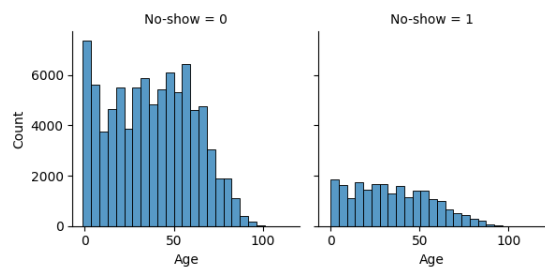
```
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.histplot(df['Age'], bins=20, kde=True, color='blue')
plt.title("Age Of Visitors")
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



```
g = sns.FacetGrid(df,col="No-show")
g.map(sns.histplot,"Age", bins=25)
plt.show()
```

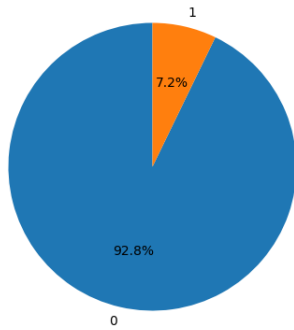


```
appet_counts = df['Diabetes'].value_counts()

plt.figure(figsize=(5,5))
plt.pie(appet_counts,
        labels=appet_counts.index,
        autopct='%1.1f%%',
        startangle=90)

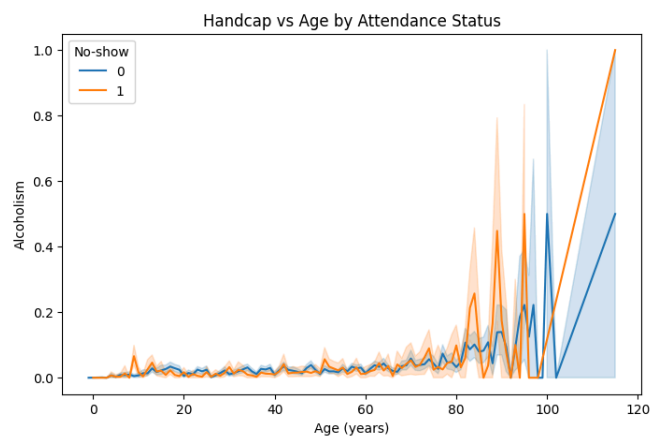
plt.title("Diabetes Status Distribution")
plt.show()
```

Diabetes Status Distribution

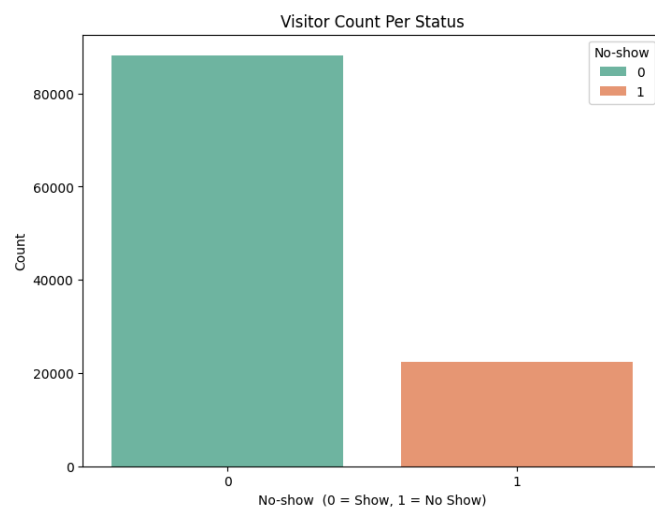


```
plt.figure(figsize=(8,5))
sns.lineplot(x='Age', y='Handcap', hue='No-show', data=df)

plt.title("Handcap vs Age by Attendance Status")
plt.xlabel("Age (years)")
plt.ylabel("Alcoholism")
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.countplot(x='No-show', data=df, hue='No-show', palette='Set2')
plt.title('Visitor Count Per Status')
plt.xlabel('No-show (0 = Show, 1 = No Show)')
plt.ylabel('Count')
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.boxplot(x='Diabetes', y='Age', data=df, palette='Set2')
plt.title('Age Distribution by Diabetes Status')
plt.xlabel('Diabetes (0 = No, 1 = Yes)')
plt.ylabel('Age')
plt.show()
```

