```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.
import kagglehub
prasad22_healthcare_dataset_path = kagglehub.dataset_download('prasad22/healthcare-dataset')

print('Data source import complete.')
```

```
Data source import complete.
```

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

```
df=pd.read_csv("/kaggle/input/healthcare-dataset/healthcare_dataset.csv")
```

```
df
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admission Type	Discl
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	Blue Cross	18856.281306	328	Urgent	202
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medicare	33643.327287	265	Emergency	201
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aetna	27955.096079	205	Emergency	202
3	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,	Medicare	37909.782410	450	Elective	202
4	adrIENNE bEll	43	Female	AB+	Cancer	2022-09-19	Kathleen Hanna	White-White	Aetna	14238.317814	458	Urgent	202
...	...	...	...	...	...	...	...	...	...	...	...	...	
55495	eLIZABeTH jaCkSOn	42	Female	O+	Asthma	2020-08-16	Joshua Jarvis	Jones-Thompson	Blue Cross	2650.714952	417	Elective	202
55496	KYle pEREz	61	Female	AB-	Obesity	2020-01-23	Taylor Sullivan	Tucker-Moyer	Cigna	31457.797307	316	Elective	202
55497	HEATher WaNG	38	Female	B+	Hypertension	2020-07-13	Joe Jacobs DVM	and Mahoney Johnson Vasquez,	UnitedHealthcare	27620.764717	347	Urgent	202
55498	JENniFER JOneS	43	Male	O-	Arthritis	2019-05-25	Kimberly Curry	Jackson Todd and Castro,	Medicare	32451.092358	321	Elective	201
55499	jAMES GARCiA	53	Female	O+	Arthritis	2024-04-02	Dennis Warren	Henry Sons and	Aetna	4010.134172	448	Urgent	202

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55500 entries, 0 to 55499
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Name                55500 non-null  object
 1   Age                 55500 non-null  int64
 2   Gender              55500 non-null  object
 3   Blood Type          55500 non-null  object
 4   Medical Condition   55500 non-null  object
 5   Date of Admission   55500 non-null  object
 6   Doctor              55500 non-null  object
 7   Hospital            55500 non-null  object
 8   Insurance Provider  55500 non-null  object
 9   Billing Amount      55500 non-null  float64
 10  Room Number         55500 non-null  int64
 11  Admission Type      55500 non-null  object
 12  Discharge Date      55500 non-null  object
 13  Medication          55500 non-null  object
 14  Test Results        55500 non-null  object
dtypes: float64(1), int64(2), object(12)
memory usage: 6.4+ MB
```

```
df['Date of Admission']=pd.to_datetime(df['Date of Admission'])
df['Discharge Date']=pd.to_datetime(df['Discharge Date'])
```

```
df.describe()
```

	Age	Date of Admission	Billing Amount	Room Number	Discharge Date
count	55500.000000	55500	55500.000000	55500.000000	55500
mean	51.539459	2021-11-01 01:02:22.443243008	25539.316097	301.134829	2021-11-16 13:15:20.821621504
min	13.000000	2019-05-08 00:00:00	-2008.492140	101.000000	2019-05-09 00:00:00
25%	35.000000	2020-07-28 00:00:00	13241.224652	202.000000	2020-08-12 00:00:00
50%	52.000000	2021-11-01 00:00:00	25538.069376	302.000000	2021-11-17 00:00:00
75%	68.000000	2023-02-03 00:00:00	37820.508436	401.000000	2023-02-18 00:00:00
max	89.000000	2024-05-07 00:00:00	52764.276736	500.000000	2024-06-06 00:00:00
std	19.602454	NaN	14211.454431	115.243069	NaN

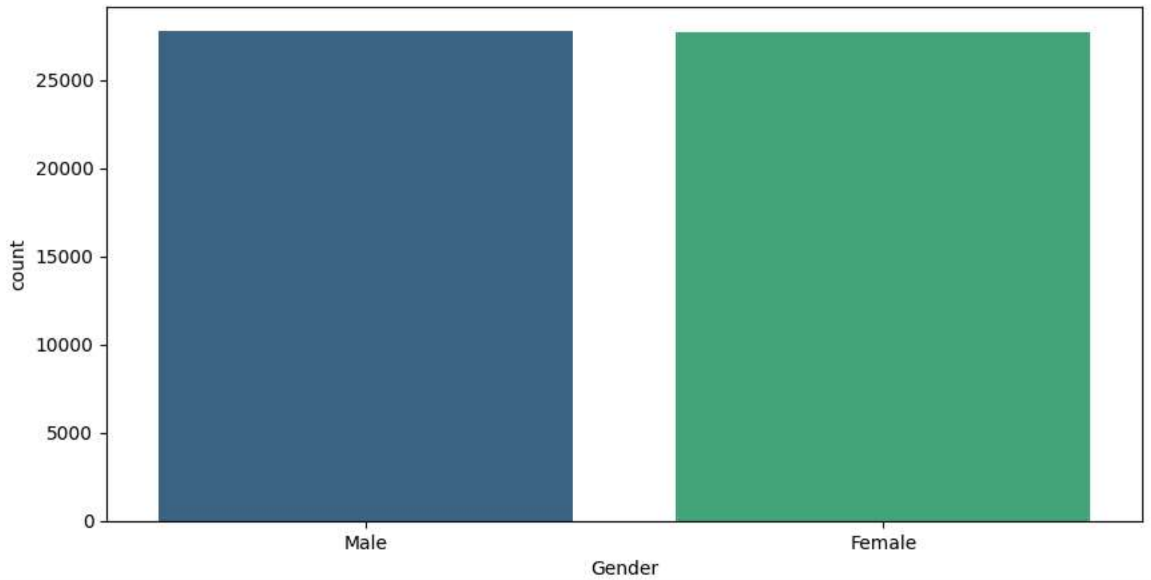
```
df.isnull().sum()
```

	0
Name	0
Age	0
Gender	0
Blood Type	0
Medical Condition	0
Date of Admission	0
Doctor	0
Hospital	0
Insurance Provider	0
Billing Amount	0
Room Number	0
Admission Type	0
Discharge Date	0
Medication	0
Test Results	0

dtype: int64

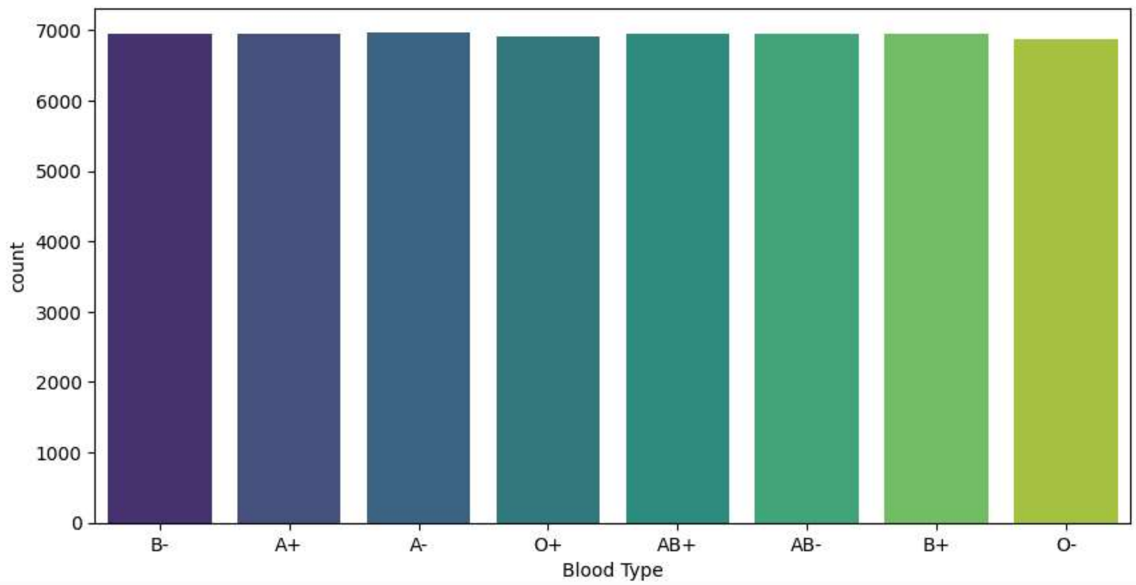
```
# df['Gender'].value_counts()
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Gender',palette='viridis')
plt.show()
```

```
<ipython-input-10-b3fd94767f69>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Gender',palette='viridis')
```



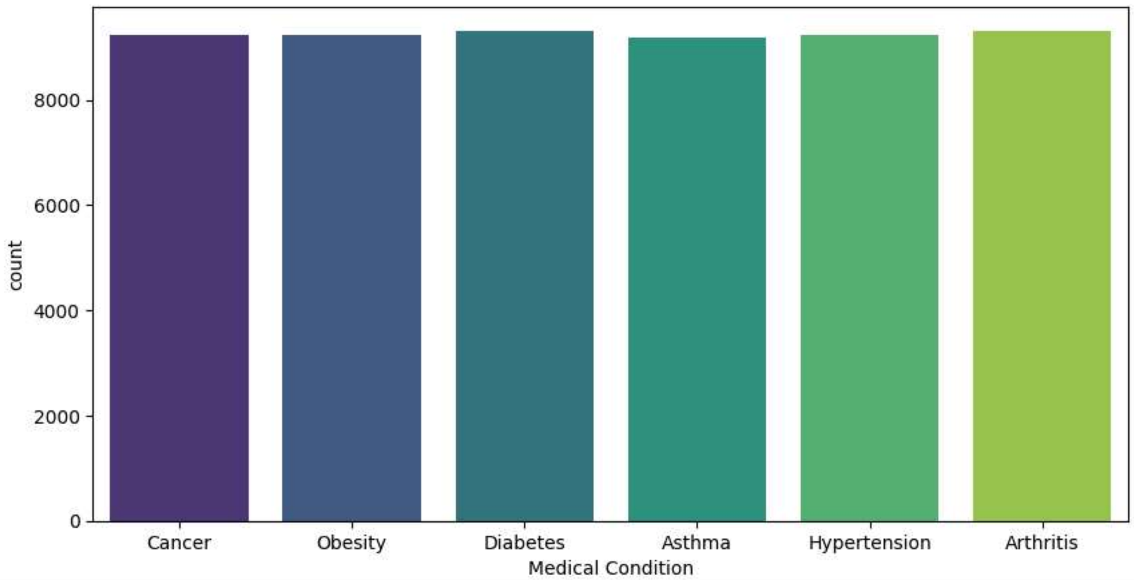
```
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Blood Type',palette='viridis')
plt.show()
```

```
<ipython-input-11-ed50d1db5196>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Blood Type',palette='viridis')
```



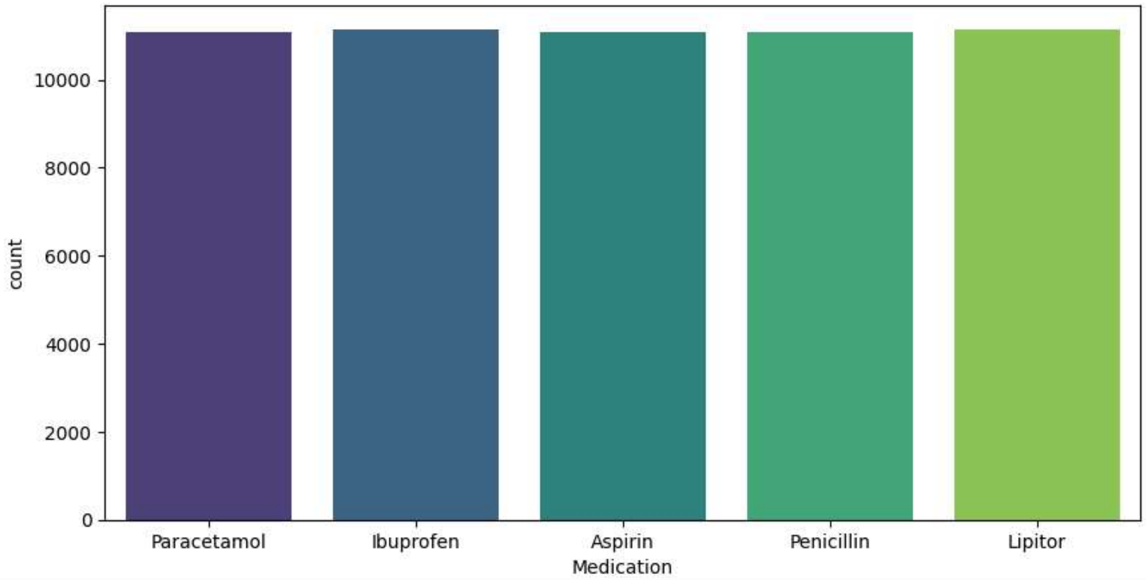
```
# df['Medical Condition'].value_counts()
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Medical Condition',palette='viridis')
plt.show()
```

```
<ipython-input-12-c4a3b704f2e4>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Medical Condition',palette='viridis')
```



```
df['Medication'].value_counts()
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Medication',palette='viridis')
plt.show()
# sns.barplot(x=df['Medication'].value_counts().index,y=df['Medication'].value_counts().values)
```

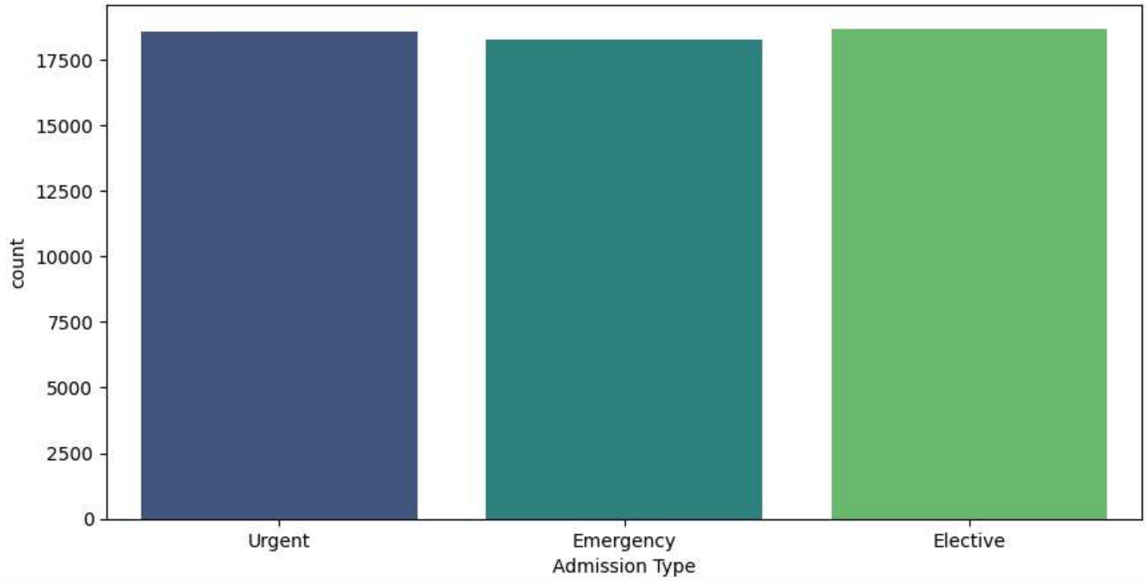
```
<ipython-input-13-1762cefd6557>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Medication',palette='viridis')
```



Start coding or generate with AI.

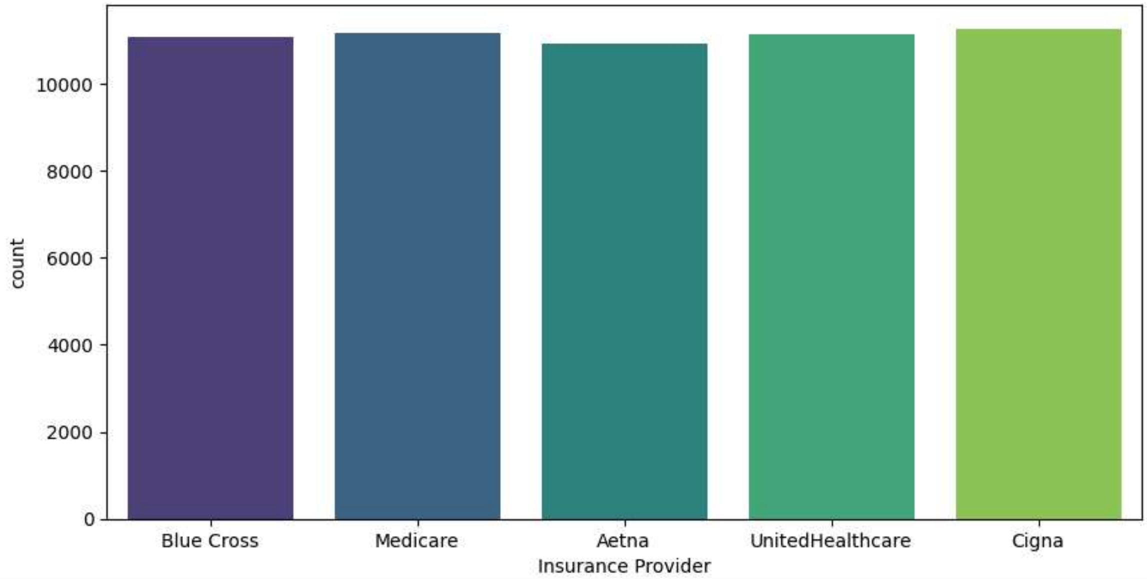
```
# df['Medical Condition'].value_counts()
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Admission Type',palette='viridis')
plt.show()
```

```
<ipython-input-14-139ffdb761f4>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Admission Type',palette='viridis')
```



```
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Insurance Provider',palette='viridis')
plt.show()
```

```
<ipython-input-15-d47ba65f04d8>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(data=df,x='Insurance Provider',palette='viridis')
```

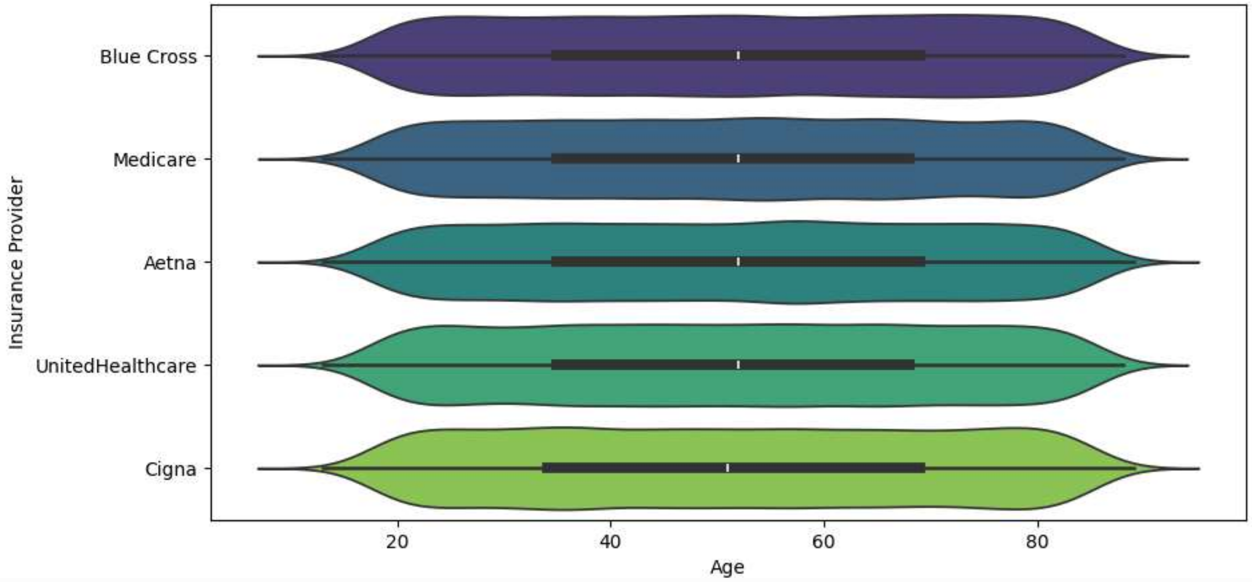


```
tdf=df.copy()
tdf
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admission Type	Discl
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	Blue Cross	18856.281306	328	Urgent	202
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medicare	33643.327287	265	Emergency	201
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aetna	27955.096079	205	Emergency	202
3	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,	Medicare	37909.782410	450	Elective	202
4	adrIENNE bEll	43	Female	AB+	Cancer	2022-09-19	Kathleen Hanna	White-White	Aetna	14238.317814	458	Urgent	202
...	...	...	...	...	...	...	...	...	...	...	...	...	
55495	eLIZABeTH jaCkSOn	42	Female	O+	Asthma	2020-08-16	Joshua Jarvis	Jones-Thompson	Blue Cross	2650.714952	417	Elective	202
55496	KYle pEREz	61	Female	AB-	Obesity	2020-01-23	Taylor Sullivan	Tucker-Moyer	Cigna	31457.797307	316	Elective	202
55497	HEATher WaNG	38	Female	B+	Hypertension	2020-07-13	Joe Jacobs DVM	and Mahoney Johnson Vasquez,	UnitedHealthcare	27620.764717	347	Urgent	202
55498	JENniFER JOneS	43	Male	O-	Arthritis	2019-05-25	Kimberly Curry	Jackson Todd and Castro,	Medicare	32451.092358	321	Elective	201
55499	jAMES GARCiA	53	Female	O+	Arthritis	2024-04-02	Dennis Warren	Henry Sons and	Aetna	4010.134172	448	Urgent	202

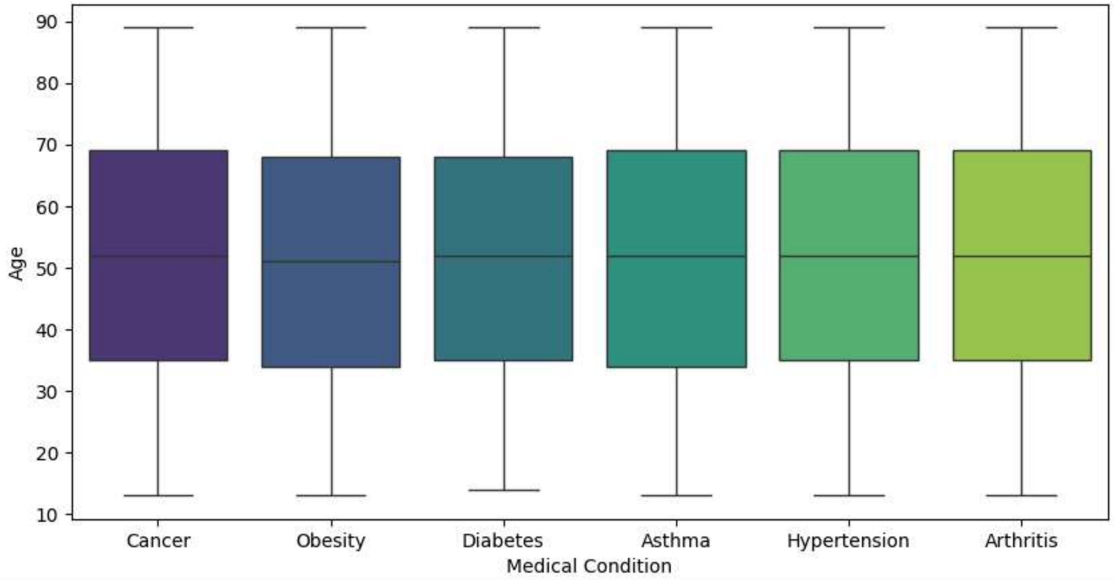
```
plt.figure(figsize=(10,5))
sns.violinplot(data=df,x='Age',y='Insurance Provider',palette='viridis')
plt.show()
```

```
<ipython-input-17-47175b17317f>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legenc

  sns.violinplot(data=df,x='Age',y='Insurance Provider',palette='viridis')
```



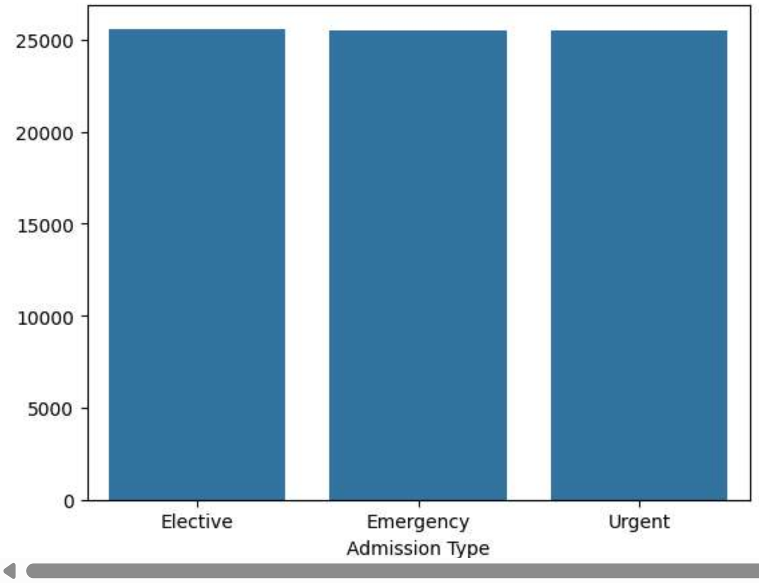
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df,x='Medical Condition',y='Age',palette='viridis')
```

```
<ipython-input-18-76836afd8744>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.boxplot(data=df,x='Medical Condition',y='Age',palette='viridis')
<Axes: xlabel='Medical Condition', ylabel='Age'>
```



```
df.groupby(['Medical Condition'])['Billing Amount'].sum()
```

```
# the average billing amount for different admission types?
admission_billing=df.groupby(['Admission Type'])['Billing Amount'].mean()
sns.barplot(x=admission_billing.index,y=admission_billing.values)
plt.show()
```



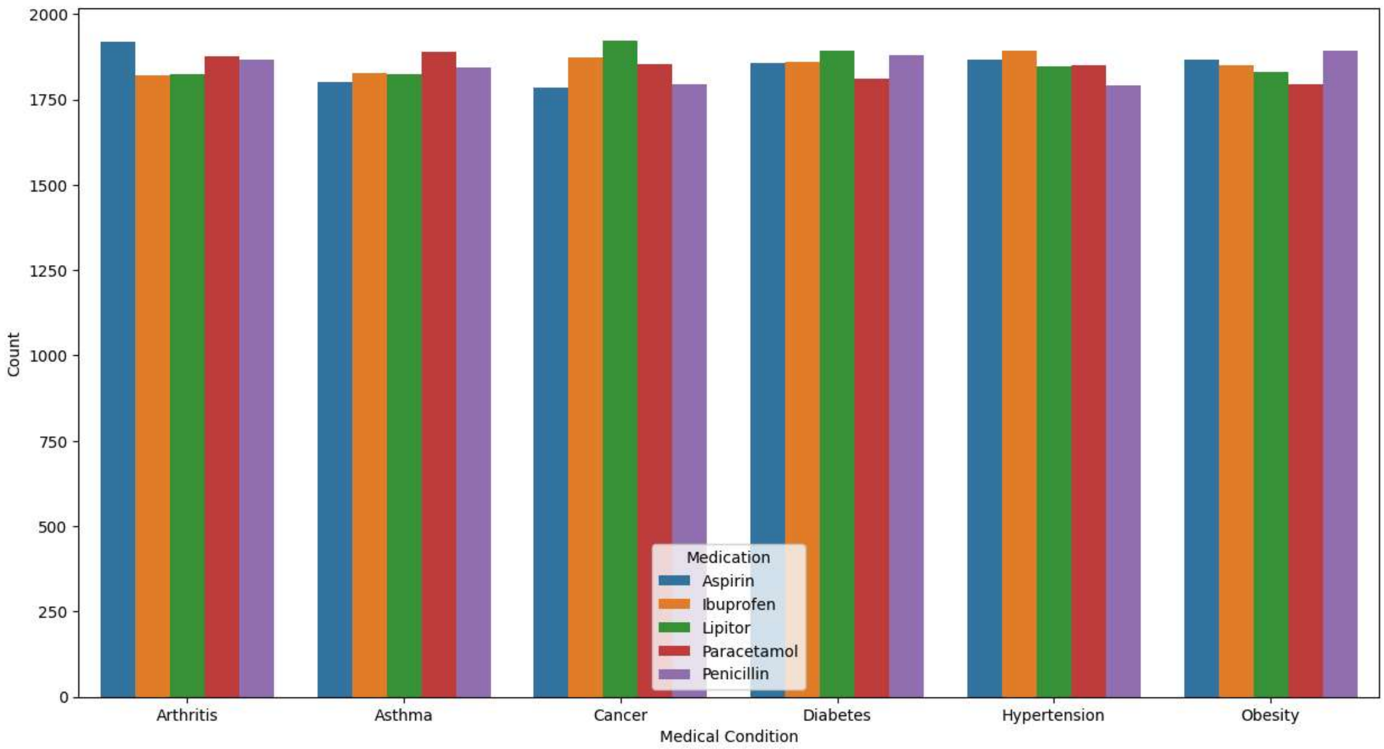
```
df['Duration of Stay']=df['Discharge Date']-df['Date of Admission']
df['Duration of Stay']=df['Duration of Stay'].dt.days
```

```
# the average duration of hospital stays for different medical conditions
df.groupby(['Medical Condition'])['Duration of Stay'].mean()
# sns.histplot(data=df,x=)
```

	Duration of Stay
Medical Condition	
Arthritis	15.517404
Asthma	15.696570
Cancer	15.495827
Diabetes	15.422936
Hypertension	15.458626
Obesity	15.464305

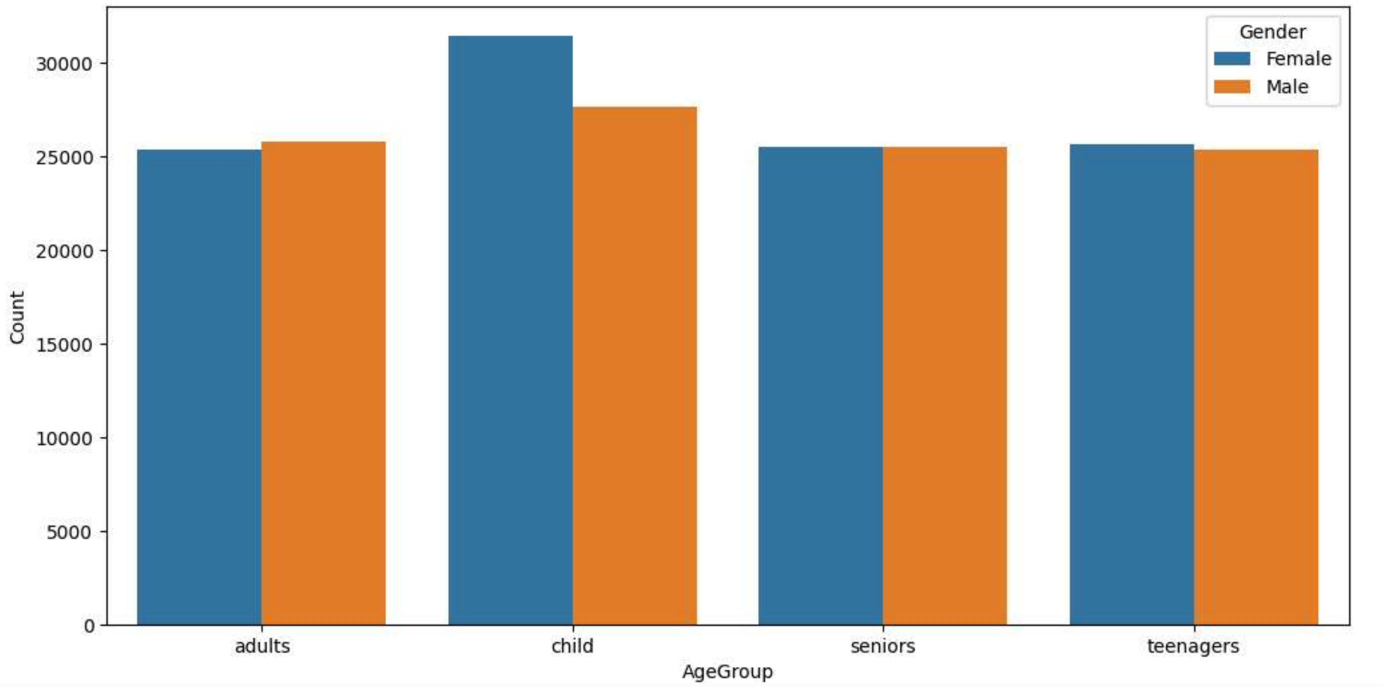
dtype: float64

```
plt.figure(figsize=(15,8))
mcm=df.groupby(['Medical Condition','Medication']).size().reset_index(name='Count')
sns.barplot(data=mcm,x='Medical Condition',y='Count',hue='Medication')
plt.show()
```

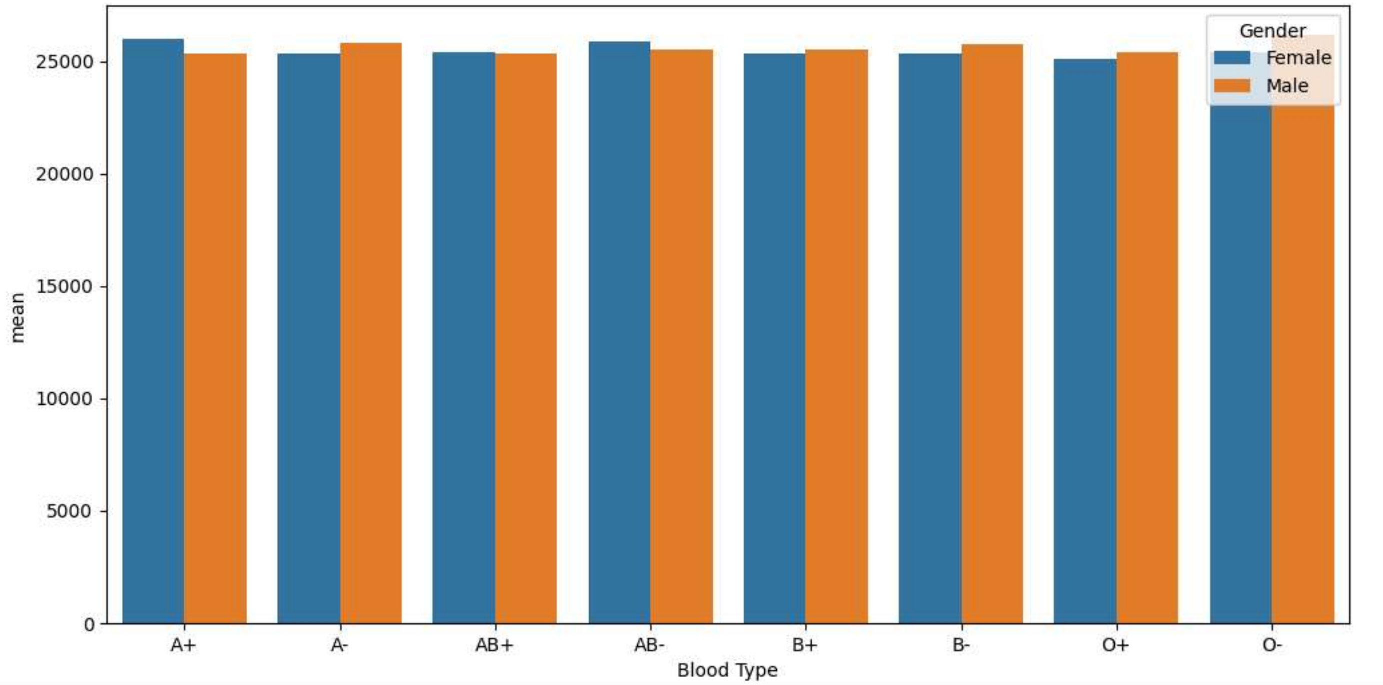


```
def a2gr(age):
  if age<14:
    return 'child'
  elif age>=14 and age <25:
    return 'teenagers'
  elif age>=25 and age <50:
    return 'adults'
  else:
    return 'seniors'

df['AgeGroup']=df['Age'].apply(lambda x: a2gr(x))
```

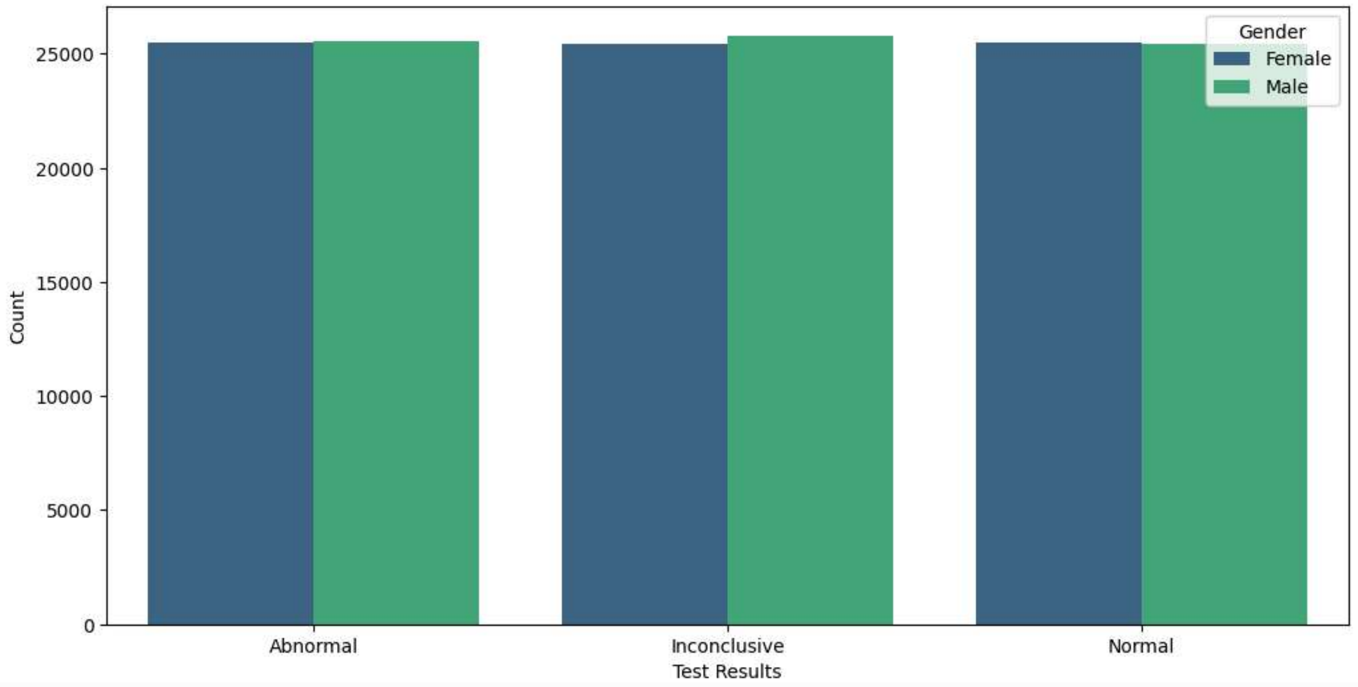
```
plt.figure(figsize=(12,6))
ABill=df.groupby(['Gender','AgeGroup'])['Billing Amount'].mean().reset_index(name='Count')
sns.barplot(data=ABill,x='AgeGroup',y='Count',hue='Gender')
plt.show()
```



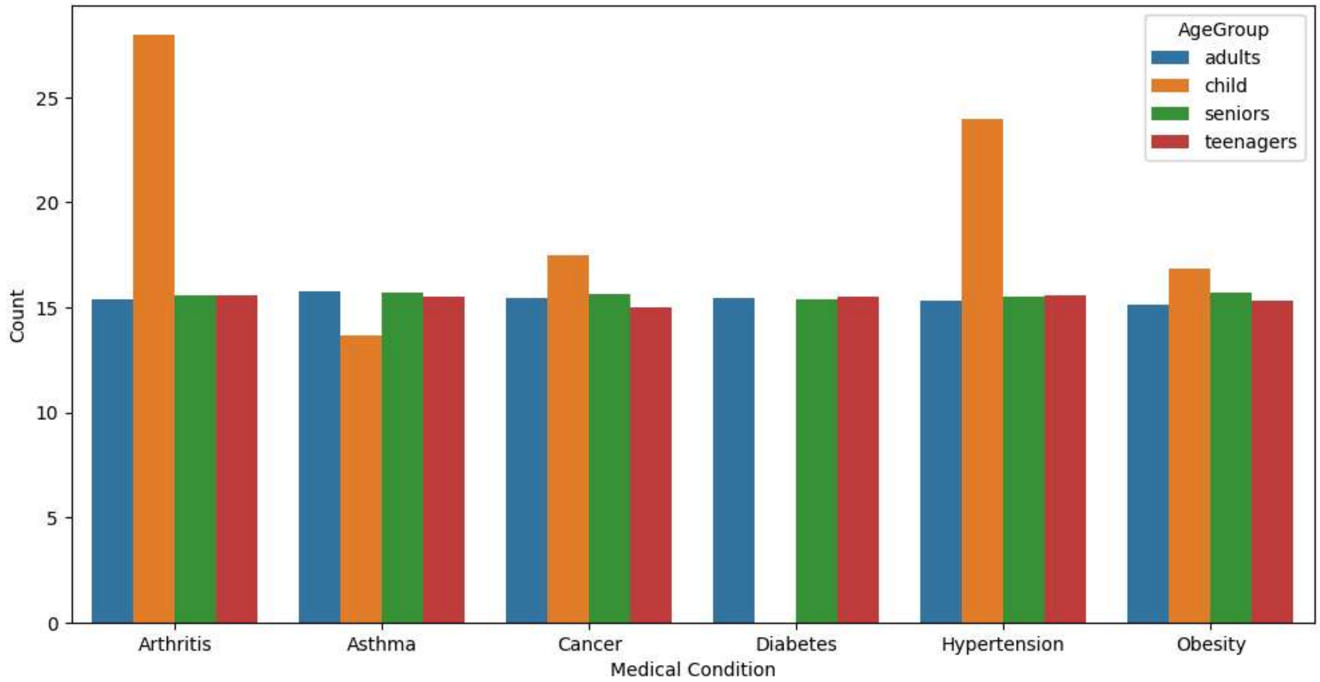
```
plt.figure(figsize=(12,6))
ABill=df.groupby(['Gender','Blood Type'])['Billing Amount'].mean().reset_index(name='mean')
sns.barplot(data=ABill,x='Blood Type',y='mean',hue='Gender')
plt.show()
```



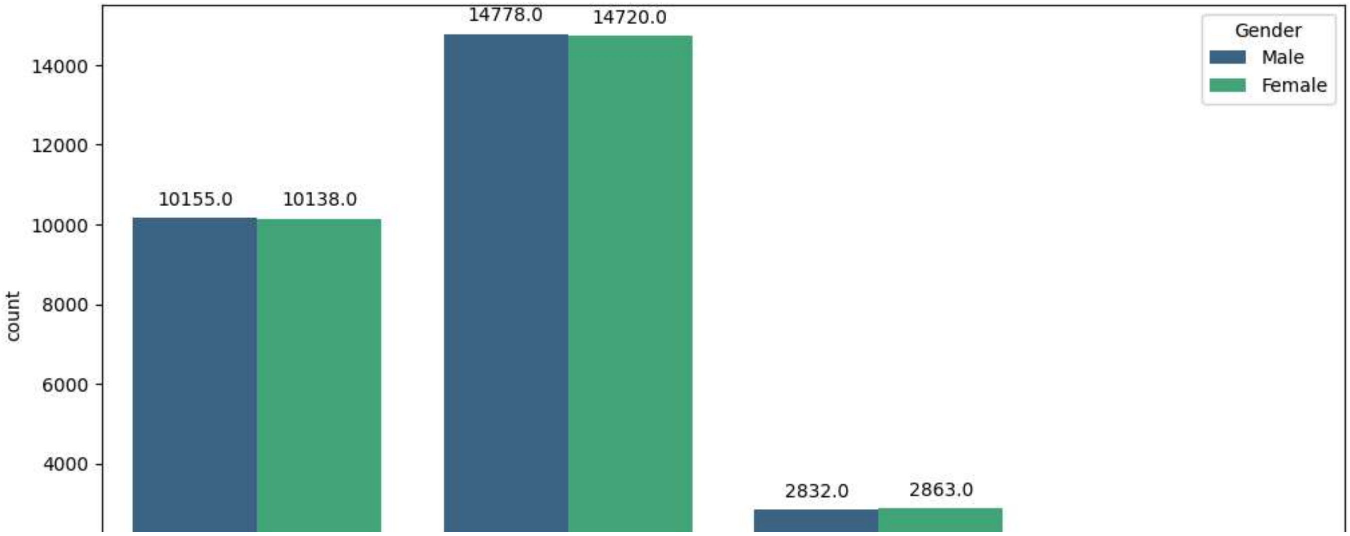
```
plt.figure(figsize=(12,6))
ABill=df.groupby(['Gender','Test Results'])['Billing Amount'].mean().reset_index(name='Count')
sns.barplot(data=ABill,x='Test Results',y='Count',hue='Gender',palette='viridis')
plt.show()
```



```
plt.figure(figsize=(12,6))
ABill=df.groupby(['Medical Condition','AgeGroup'])['Duration of Stay'].mean().reset_index(name='Count')
sns.barplot(data=ABill,x='Medical Condition',y='Count',hue='AgeGroup')
plt.show()
```



```
# sns.countplot(data=df,x='AgeGroup',hue='Gender',palette='viridis')
plt.figure(figsize=(12,6))
ax = sns.countplot(data=df, x='AgeGroup', hue='Gender', palette='viridis')

for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2, p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.show()
```


14000
12000
10000
8000
6000
4000
count
10155.0
10138.0
14778.0
14720.0
2832.0
2863.0
Gender
Male
Female