DSC630 Week1 - Assignment

Guruprasad Velikadu Krishnamoorthy

In [1]:
```
# Importing the Required libraries
import pandas as pd
import numpy as np
import os
import sys
import re
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
```

In [2]:
```
# Setting global options for the notebook such as maxrows
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_colwidth', None)
pd.set_option("display.max_rows", 100)
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# Importing the Dataset
path=os.getcwd()
# Assigning a path for the file
# Source of the dataset: https://www.kaggle.com/datasets/prasad22/healthcare-dataset/data
health_file_path=path+"\\healthcare_dataset.csv"
```

In [4]:
```
# Loading the source file into Pandas DataFrame
health_df_orig=pd.read_csv(health_file_path)
# Printing the shape of the dataframe
health_df_orig.shape
```

Out[4]: (10000, 15)

In [5]:
```
hosp_df=health_df_orig.copy()
```

```
# Printing top 5 rows of the Dataframe
hosp_df=hosp_df.sample(3000, random_state=28)
hosp_df.head()
```

Out[5]:

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Ad
4345	Jessica Werner	36	Female	AB+	Obesity	2019-02-08	Derek Perkins	Evans, Ruiz and Wood	Blue Cross	47705.167028	265	
5182	Kendra Walker	48	Female	A-	Asthma	2023-07-10	Veronica Webb	Wells Ltd	Aetna	14897.824044	329	
1597	Renee Walker	78	Female	A+	Diabetes	2021-04-17	Troy Harmon	Henderson, Mercado and Webb	Cigna	10594.497573	222	
4459	Natalie Bradley	58	Female	B+	Cancer	2019-09-22	Nicholas Rodriguez	Hernandez LLC	UnitedHealthcare	12963.365016	491	
4168	Cynthia Mclean	85	Male	O-	Obesity	2020-03-11	Taylor Gardner	Burke-Mendoza	Cigna	1889.902251	201	

In [6]:
```
# Getting statitical summary of the dataset
hosp_df.describe()
```

Out[6]:

	Age	Billing Amount	Room Number
count	3000.000000	3000.000000	3000.000000
mean	51.541000	25163.451909	298.453667
std	19.626565	14011.953683	117.566826
min	18.000000	1020.337790	101.000000
25%	35.000000	13128.029684	195.000000
50%	52.000000	24739.950418	298.000000
75%	69.000000	37359.468670	398.250000
max	85.000000	49985.973068	500.000000

In [7]:
```
# Getting statistical summary of the dataset
hosp_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3000 entries, 4345 to 1915
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                3000 non-null   object 
 1   Age                 3000 non-null   int64  
 2   Gender              3000 non-null   object 
 3   Blood Type          3000 non-null   object 
 4   Medical Condition   3000 non-null   object 
 5   Date of Admission   3000 non-null   object 
 6   Doctor              3000 non-null   object 
 7   Hospital            3000 non-null   object 
 8   Insurance Provider  3000 non-null   object 
 9   Billing Amount      3000 non-null   float64
 10  Room Number         3000 non-null   int64  
 11  Admission Type      3000 non-null   object 
 12  Discharge Date      3000 non-null   object 
 13  Medication          3000 non-null   object 
 14  Test Results        3000 non-null   object 
dtypes: float64(1), int64(2), object(12)
memory usage: 375.0+ KB
```

1. Write a summary of your data and identify at least two questions to explore visually with your data

1.1. Summary: The Dataset contains healthcare related data that is taken from Kaggle (https://www.kaggle.com/datasets/prasad22/healthcare-dataset/data) that contains the data created for educational and research purposes. The dataset contains 10000 rows and 15 columns.

From the initial obsevation, it appears the column 'room number' is loaded as Integer which should be converted to Object type and the date of Admission and Discharge to Date types. The datatype of the rest of columns are loaded as expected.

Below is the description of some of the columns that will be explored in this assignment.

1. Age: Represents the age of the patient and is expressed in numbers.

2. Gender: Represents the gender of the patient

3. Blood Type: Represents the Blood type of the patient.

4. Medical Condition: Represents the health condition or diagnosis of the patient.

5. Date of admission: Represents the date when the patient was admitted to the hospital.

6. Billing Amount: Amount in USD billed during the patient stay or treatment for the diagnosis.

7. Discharge date: Date when the patient was discharged from the hospital.

1.2. Questions to Explore:

i. What is the common health condition in both male and female patients aged 65 and above?

ii. During which month of the year were more patients treated for Diabetes?

iii.What is the average amount billed for treating in Emergency vs Urgent Care?

iv. Is there any pattern in the Patient stays in hospitals for cancer treatment?

1.3. Checking for nulls:

In [8]:
```
# Getting the nulls in each column using the isna function
hosp_df.isna().sum()
```

Out[8]:
```
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
```

1.4. Performing Data type conversions:

In [9]:
```
# Converting the Room Number columns as Object type
hosp_df['Room_Number']=hosp_df['Room Number'].astype("object")
```

In [10]:
```
# Converting string back to date
def convert_string_to_date(input_str):
    """
    This function converts string back to date using strptime method
    """
    return datetime.strptime(input_str,"%Y-%m-%d")
# Applying the function to Admission and Discharge data to convert them to Date datatype
hosp_df["Date_of_Admission"]=hosp_df["Date of Admission"].apply(convert_string_to_date)
hosp_df["Discharge_Date"]=hosp_df["Discharge Date"].apply(convert_string_to_date)
```

In [11]:
```
# Adding a new column to represent the number of hospital stays
hosp_df["hospital_stay"]=(hosp_df["Discharge_Date"]-hosp_df["Date_of_Admission"])/np.timedelta64(1, 'D')
# Converting the number of hospital stays to Integer
hosp_df["hospital_stay"]=hosp_df["hospital_stay"].astype(int)
```

In [12]:
```
# Creating new columns for Year and Month of Admission
hosp_df["year_of_admission"]=hosp_df["Date_of_Admission"].dt.year
hosp_df["month_of_admission"]=hosp_df["Date_of_Admission"].dt.month
```

In [13]:
```
# Printing the top 5 rows from the dataframe
hosp_df.head()
```

Out[13]:

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Ad
4345	Jessica Werner	36	Female	AB+	Obesity	2019-02-08	Derek Perkins	Evans, Ruiz and Wood	Blue Cross	47705.167028	265	
5182	Kendra Walker	48	Female	A-	Asthma	2023-07-10	Veronica Webb	Wells Ltd	Aetna	14897.824044	329	
1597	Renee Walker	78	Female	A+	Diabetes	2021-04-17	Troy Harmon	Henderson, Mercado and Webb	Cigna	10594.497573	222	
4459	Natalie Bradley	58	Female	B+	Cancer	2019-09-22	Nicholas Rodriguez	Hernandez LLC	UnitedHealthcare	12963.365016	491	
4168	Cynthia Mclean	85	Male	O-	Obesity	2020-03-11	Taylor Gardner	Burke-Mendoza	Cigna	1889.902251	201	

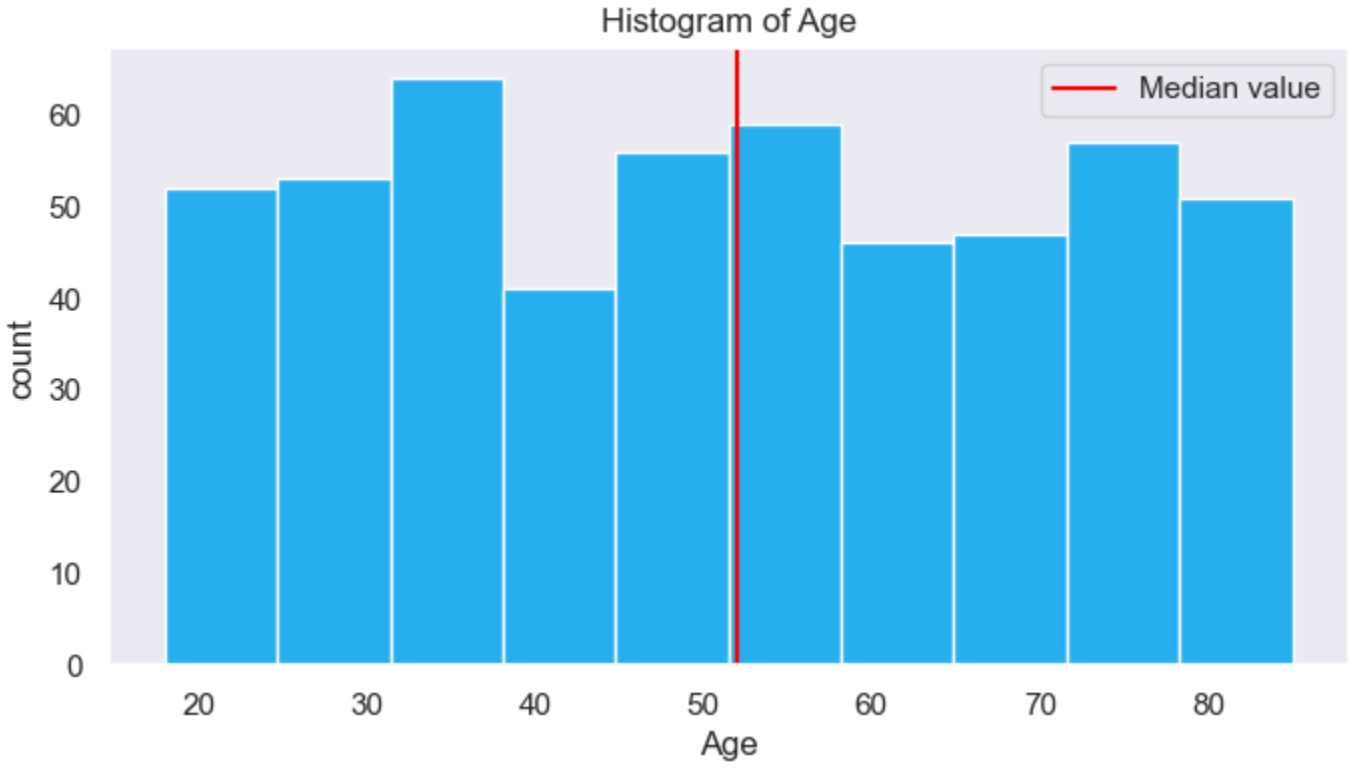
2. Create a histogram or bar graph from your data

In [14]:
```
# Creating a Dataframe that contains data of Asthma patients
asthma_df=hosp_df[hosp_df["Medical Condition"].isin(["Asthma"])]
```

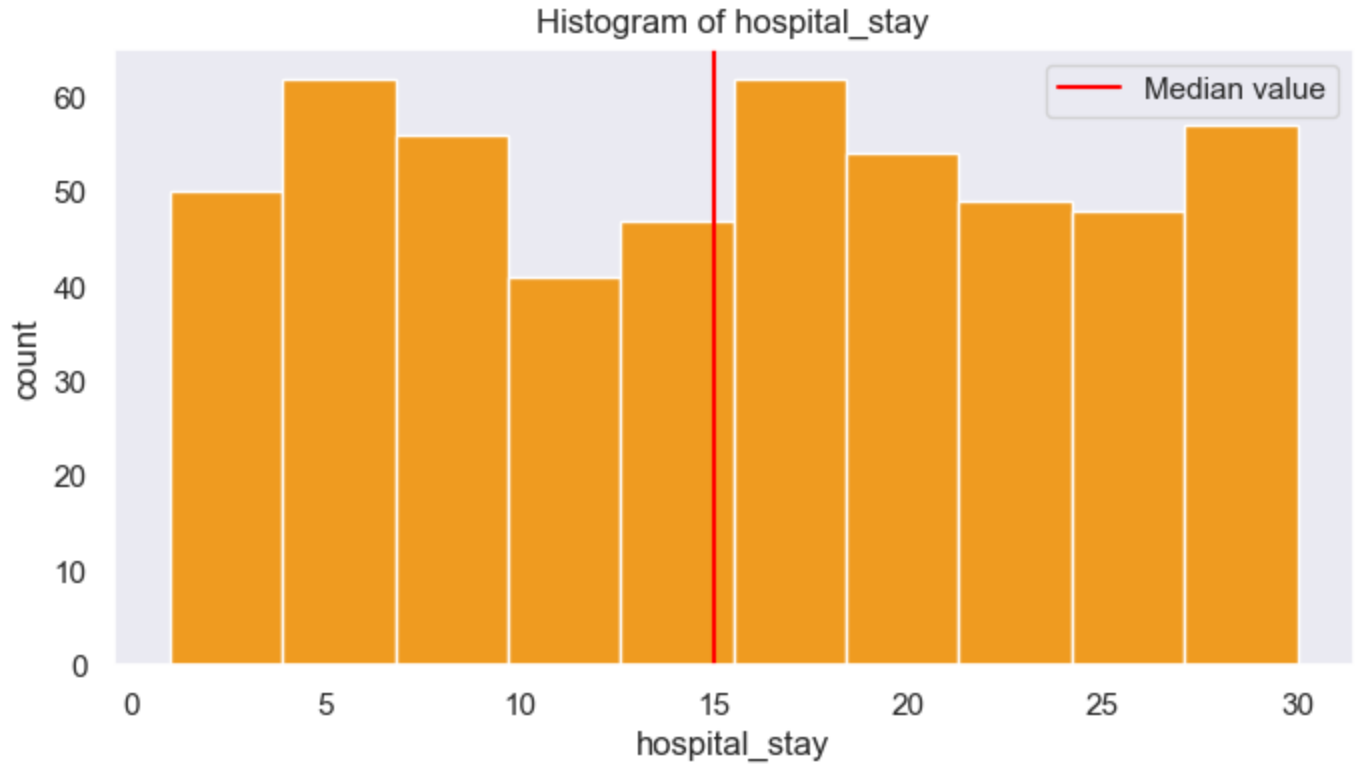
In [15]:
```
# Getting the list of Numeric columns that should be represented in the plot
selected_cols=["Age","hospital_stay","Billing Amount"]
sns.set(font_scale=1)
# Setting the colors to be used in the plot
colors=[ "#27aeef", "#ef9b20", "#87bc45"]
# Enumerating through the columns and creating Histogram for each column
for inx,col in enumerate(selected_cols):
    # Computing the median value
    median_val = hosp_df[col].median()
    plt.figure(figsize = (8, 4))
    color = 'Red'
    # Plotting the Median line
    print(f"The Median of {col} is {median_val}")
    plt.axvline(median_val, color=color,label='Median value')
    asthma_df[col].hist(grid=False,color= colors[inx])
    # Plotting the title and the labels
```

```
    plt.xlabel(asthma_df[col].name)
    plt.ylabel('count')
    plt.legend()
    plt.title('Histogram of '+ col)
    plt.show()
```

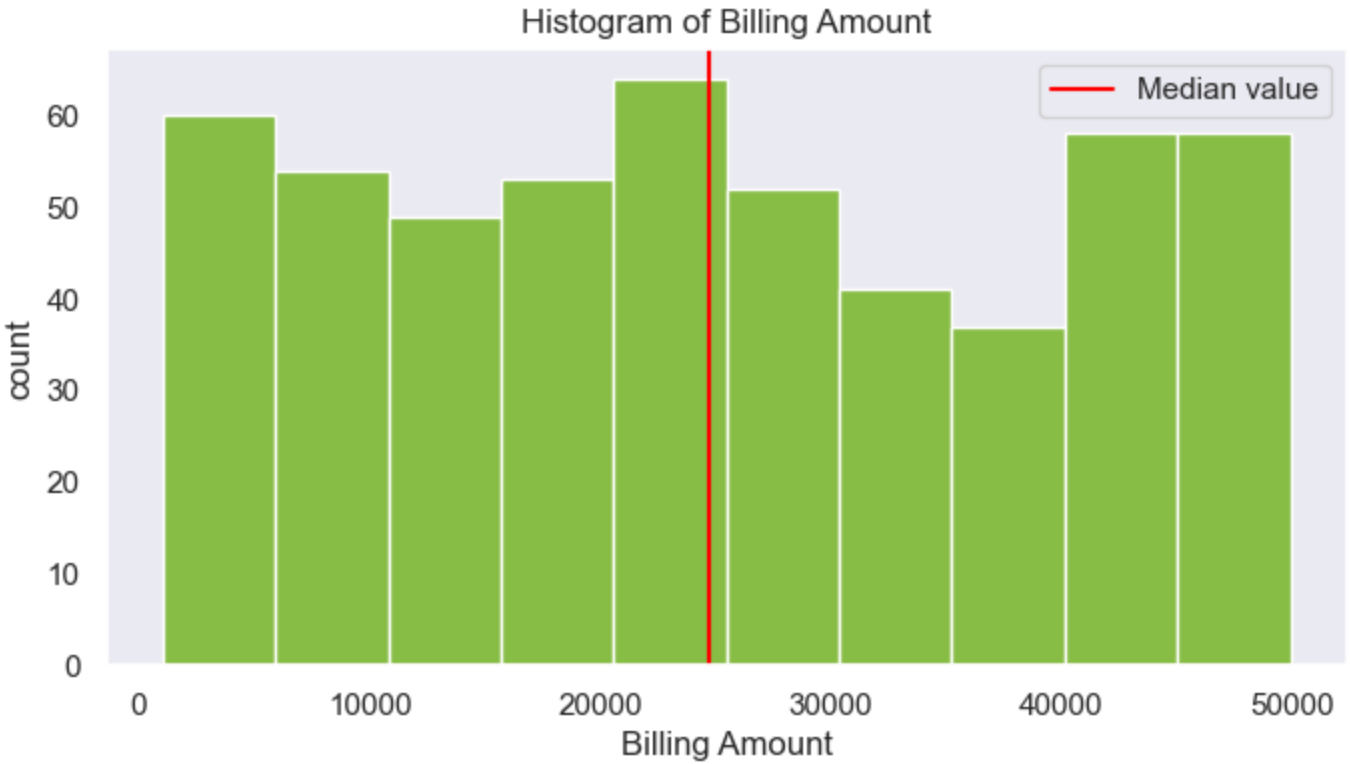
```
The Median of Age is 52.0
```




```
The Median of hospital_stay is 15.0
```



The Median of Billing Amount is 24739.950418180717



Summary of Histograms: The Histogram plots indicate that the distribution is not normal for the Age of patients, Hospital stay(in days) and the billing amount. The Median values are highlighed by the Red axisline in the plot. The Median values are also printed above each Histogram plot.

2.1. Creating a BarPlot:

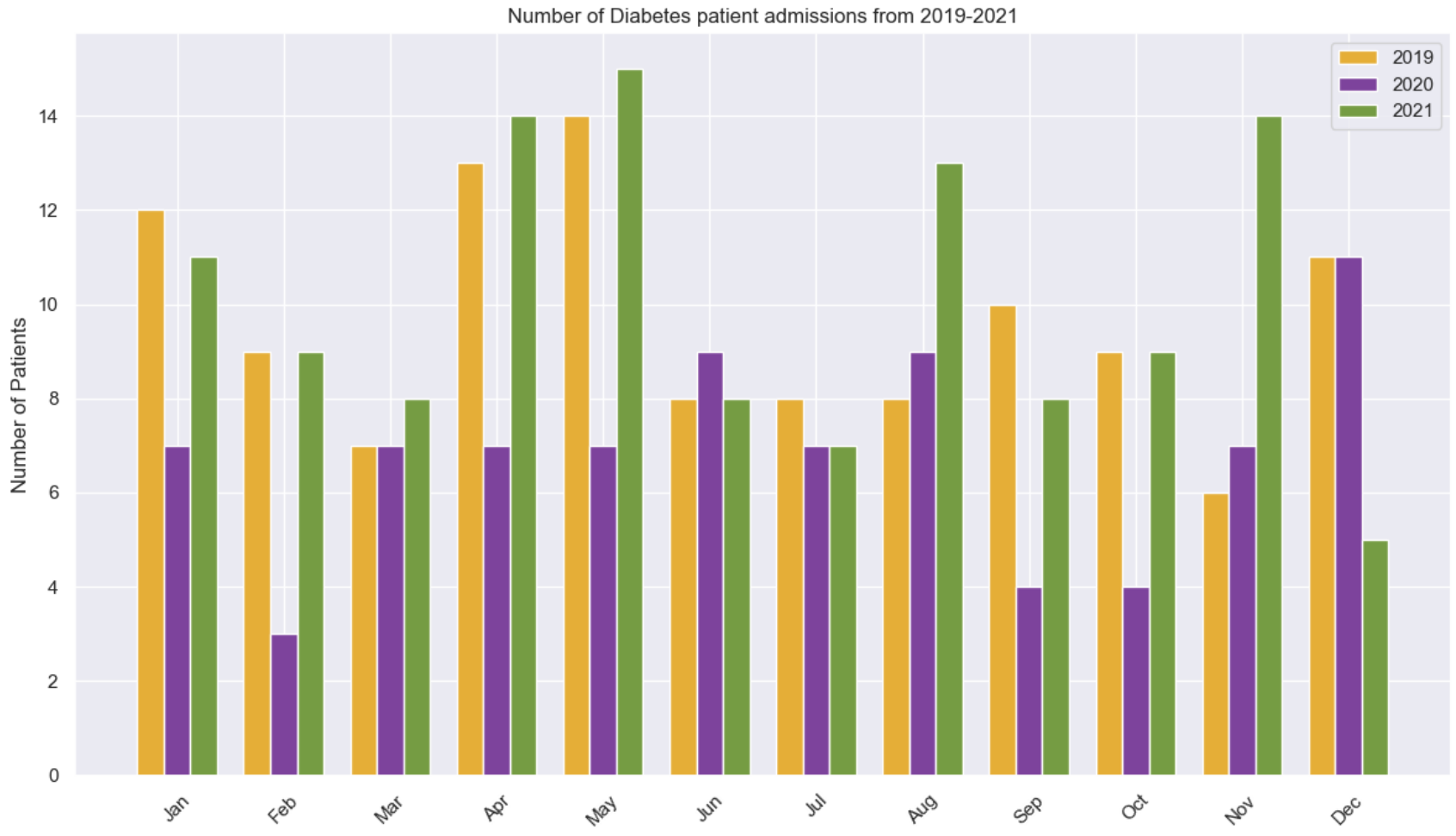
In [16]:
```
# Creating a dataframe that contains Diabetes patients
diabetes_df=hosp_df[hosp_df["Medical Condition"].isin(["Diabetes"])]
# Filtering the Diabetes data fr the years 2019-2021
diabetes_df1=asthma_df[asthma_df["year_of_admission"].isin([2019,2020,2021])]
# Grouping the Diabetes data based on the Year and Month of admission and getting the counts
diabetes_df1=diabetes_df1.groupby(["year_of_admission","month_of_admission"]).count()["Name"].reset_index()
diabetes_df1.head()
```

Out[16]:

	year_of_admission	month_of_admission	Name
0	2019	1	12
1	2019	2	9
2	2019	3	7
3	2019	4	13
4	2019	5	14

In [17]:
```
# Creating dataframes for each year to represent them in the Bar plot
diabetes_df_2019=diabetes_df1[diabetes_df1["year_of_admission"].isin([2019])]
diabetes_df_2020=diabetes_df1[diabetes_df1["year_of_admission"].isin([2020])]
diabetes_df_2021=diabetes_df1[diabetes_df1["year_of_admission"].isin([2021])]
# Creating x indexes with same number of elements as Year. This will be used in the plotting of x axis
x_indexes=np.arange(len(diabetes_df_2019["year_of_admission"]))
# Creating a list of months which will be used as labels for Xaxis
x_months=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
```

In [18]:
```
# Assigning default width for each bar in the Barchart
width=0.25
plt.figure(figsize = (12, 7))
# Plotting BarChart for each year. The Value of the Xindexes are adjusted by using the Width
plt.bar(x_indexes-width,diabetes_df_2019["Name"],width=width,color="#e5ae37",label="2019")
plt.bar(x_indexes,diabetes_df_2020["Name"],width=width,color="#7D439C",label="2020")
# The xindexes are adjusted according to the width of the Bar chart
plt.bar(x_indexes+width,diabetes_df_2021["Name"],width=width,color="#759C43",label="2021")
plt.legend()
# The x-ticks will have values of x_indexes with label values as Month.
plt.xticks(ticks=x_indexes,labels=x_months)
# Setting the Title, X and Y labels
plt.xticks(rotation=45)
plt.title("Number of Diabetes patient admissions from 2019-2021")
plt.ylabel("Number of Patients")
plt.tight_layout()
plt.show()
```

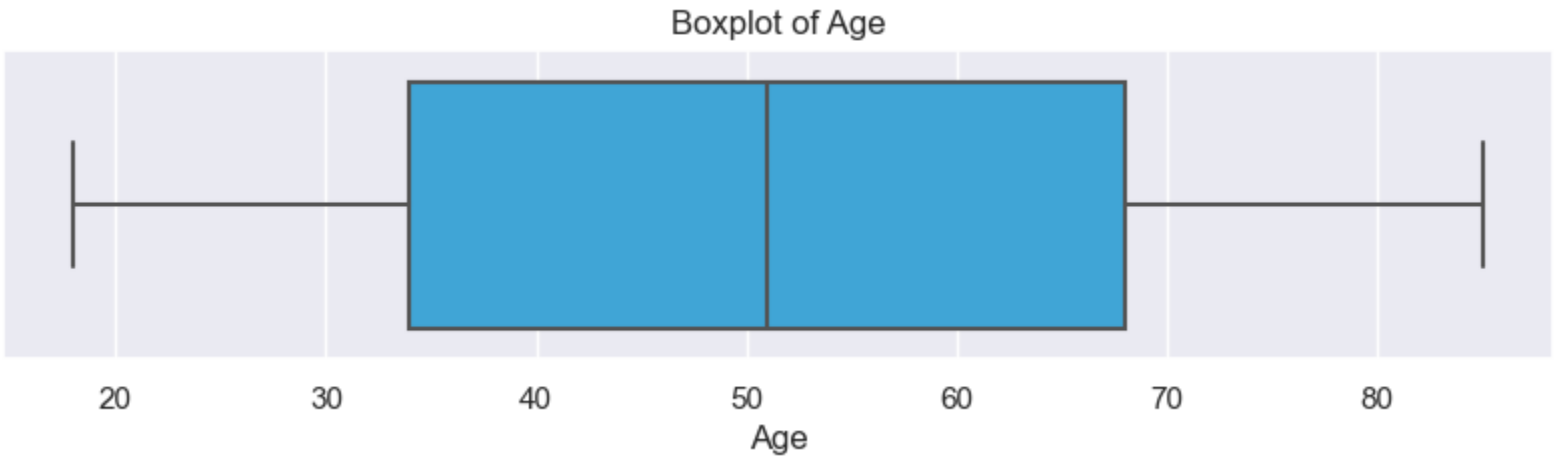


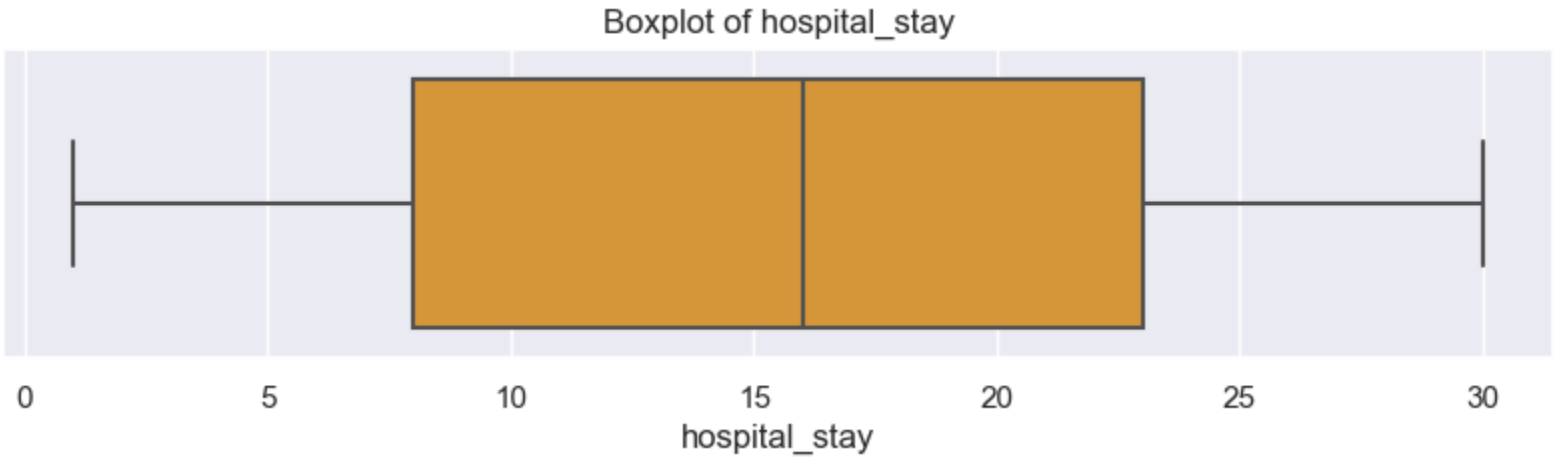
Summary of the Barplot: The Barplot indicates that more diabetes patients were admitted in the months of April and May in the years 2019 and 2021, but the number was low in 2020. In 2021, the number of Diabetes admissions on average went up compraed to the previous years.

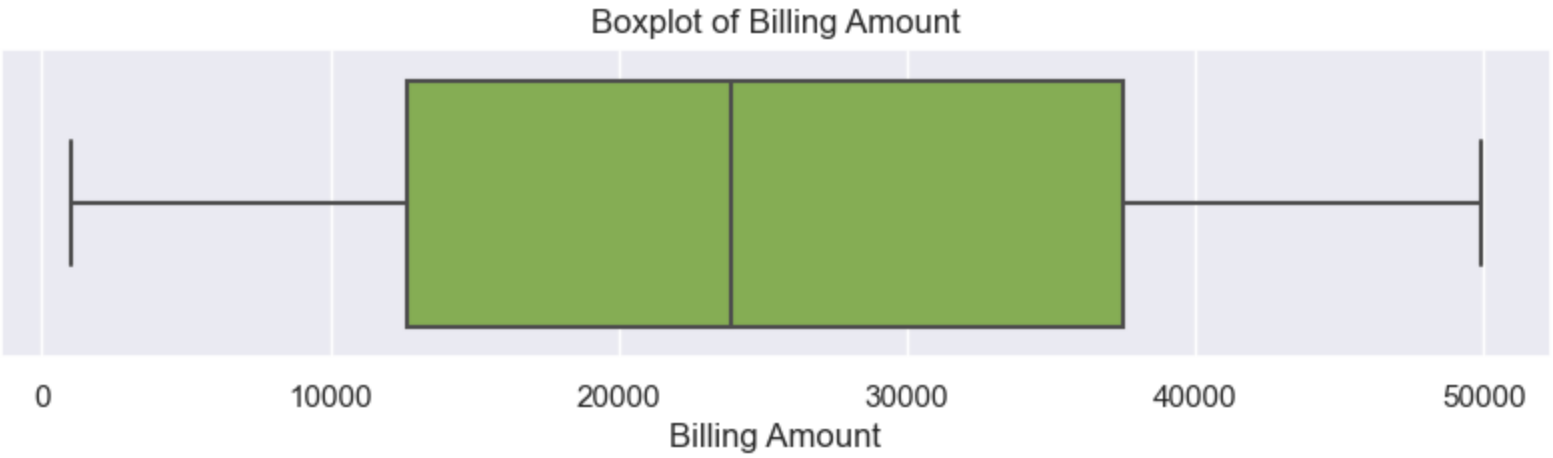
3.Create a boxplot from your data.

In [19]:
```
# Looping through each column in selected columns list and creating Box plot for each
for inx,col in enumerate(selected_cols):
    # Setting the Title of the plot
    plt.figure(figsize = (10,2))
```

```
    plt.title('Boxplot of '+ col)
    # creating boxplot
    sns.boxplot(x=asthma_df[col],color= colors[inx])
    plt.show()
```







Summary of the Boxplot: The boxplot indicates that there are no outliers in the columns AGes, Hospital stay and Bill amount, which indicates the data is relatively cleaner.The Median values are similar to what was found in the Histograms.

4. Create a bivariate plot from your data.

4.1. Line Plot

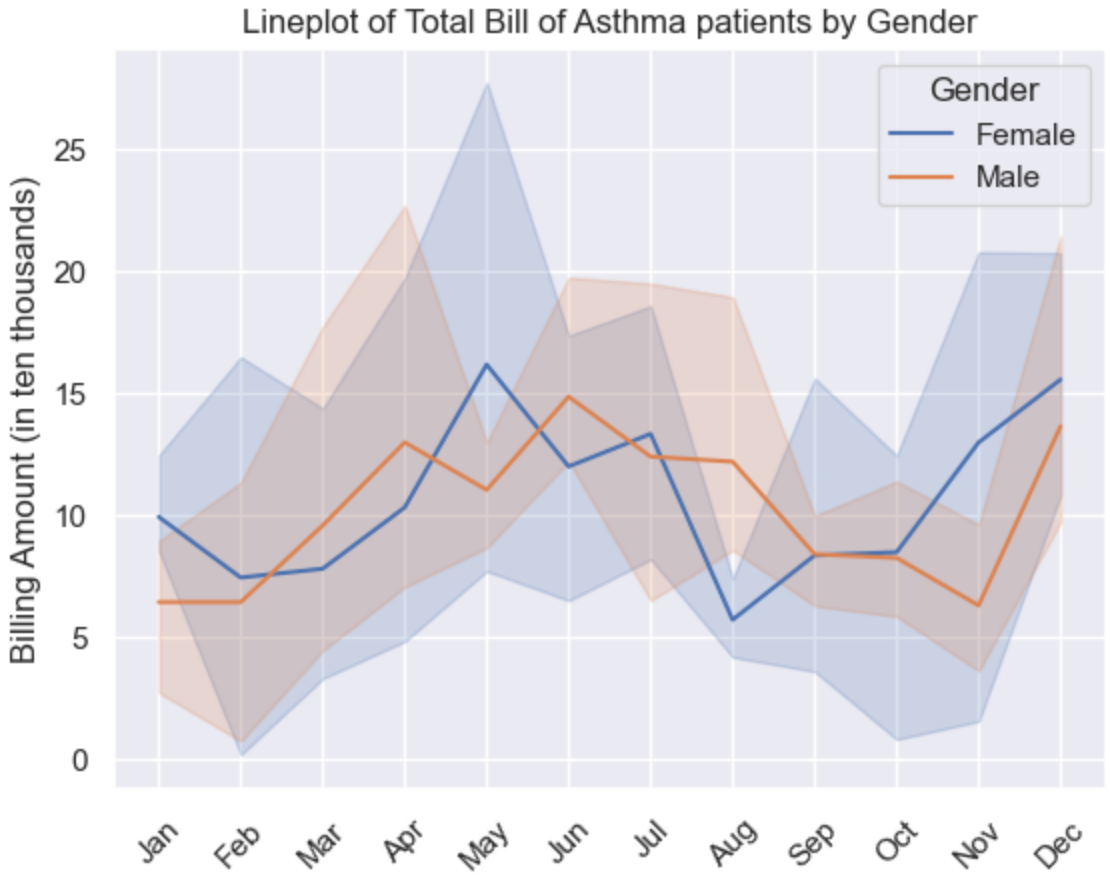
In [20]:
```
# Filtering the years in Asthma dataframe to 2019-2021
asthma_df2=asthma_df[asthma_df["year_of_admission"].isin([2019,2020,2022])]
# Grouping the Asthma data by year, month and Gender details and computing the sum of Billed amount
asthma_df2=asthma_df2.groupby(["year_of_admission","month_of_admission","Gender"]).sum()["Billing
Amount"].reset_index()
# For easy representation, dividing the Billed amount by 10000
asthma_df2["Billing Amount"]=asthma_df2["Billing Amount"]/10000
asthma_df2.head()
```

Out[20]:

	year_of_admission	month_of_admission	Gender	Billing Amount
0	2019	1	Female	12.454623
1	2019	1	Male	8.937689
2	2019	2	Female	16.485946
3	2019	2	Male	11.342729
4	2019	3	Female	5.702262

In [21]:

```
# Creating a Line plot of Bill amount of Asthma Patients by Gender for each month of year
sns.lineplot(data = asthma_df2, x = 'month_of_admission', y = 'Billing Amount', hue = 'Gender')
# Setting the title and labels
plt.title("Lineplot of Total Bill of Asthma patients by Gender")
plt.ylabel("Billing Amount (in ten thousands)")
plt.xlabel("")
# Changing the xticks and labels
plt.xticks(ticks=x_indexes+1,labels=x_months)
plt.xticks(rotation=45)
plt.figure(figsize = (8, 6))
plt.show()
```



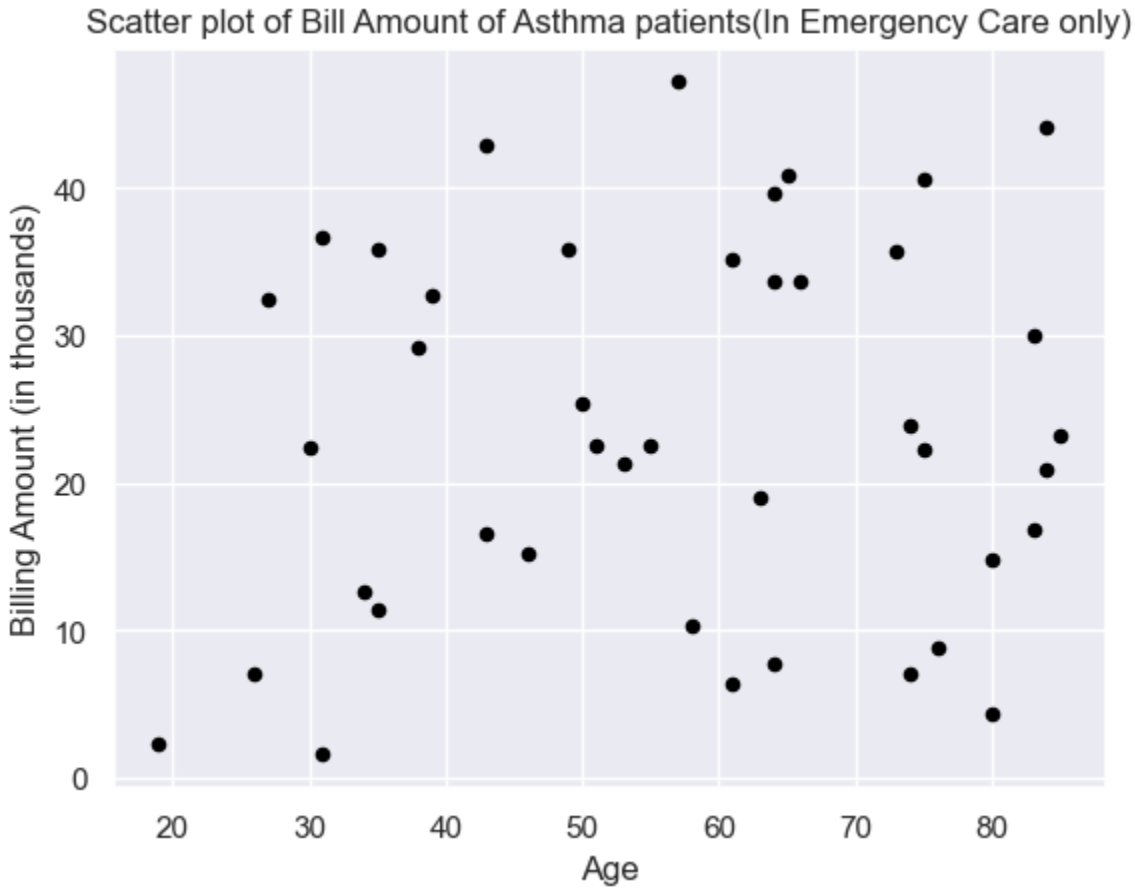

```
<Figure size 800x600 with 0 Axes>
```

Summary of the Lineplot: The Lineplot of Asthma treatments across each month of the year indicates that during May and December there were many Female patients and June had many Male patients. The treatment costs varies between 6k to 17k for treating Asthma.

4.2. Scatter Plot:

In [22]:
```
# Creating a scatter plot of Bill amount of Asthma patients that were admitted to Emergency Care in Aetna
asthma_emer_df=asthma_df[(asthma_df["Admission Type"].isin(["Emergency"])) & (asthma_df["Insurance
Provider"].isin(["Aetna"]))]
# Dividing the bill amount by 1000 for each representation
asthma_emer_df["Billing Amount"]=asthma_emer_df["Billing Amount"]/1000
# Plotting Scatter plot
```

```
asthma_emer_df.plot( x = 'Age', y = 'Billing Amount', kind = 'scatter', color='black')
# Setting the Title and Labels
plt.title("Scatter plot of Bill Amount of Asthma patients(In Emergency Care only)")
plt.ylabel("Billing Amount (in thousands)")
plt.xlabel("Age")
plt.figure(figsize = (10, 6))
plt.show()
```

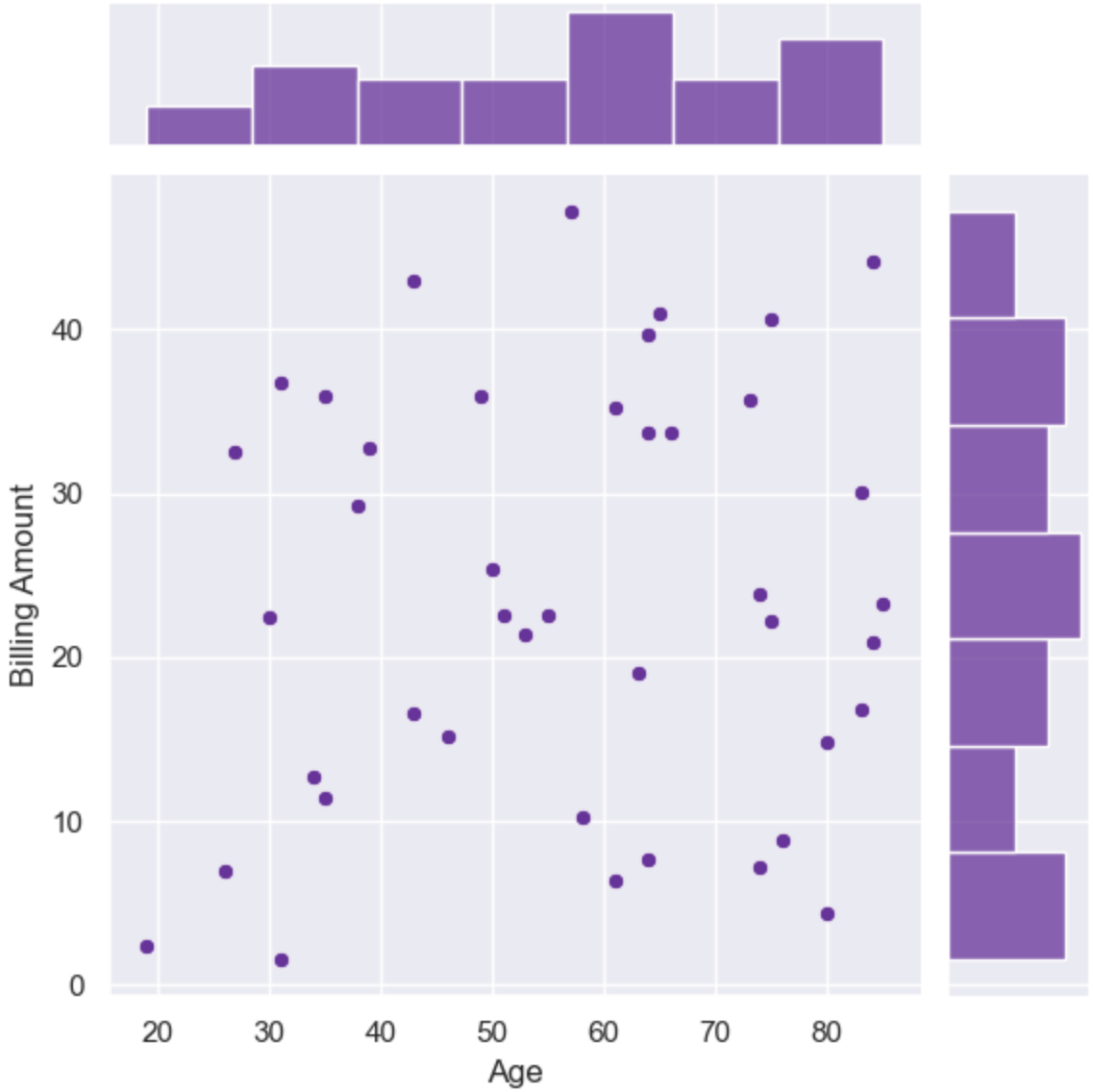



```
<Figure size 1000x600 with 0 Axes>
```

Summary of the Scatter plot: The scatter plot of Asthma treatment in emergency care does not have any distinguishable pattern. However, there were fewer patients of Ages 30 and lower. The treatment costs in emergency ranged from a few thousands upto 50k.

4.3. Joint Plot:

In [23]:
```
# Creating a Jointplot of Age vs Bill amount of the Asthma Patients
sns.jointplot(data = asthma_emer_df, x = 'Age', y = 'Billing Amount',color='rebeccapurple')
# Setting the labels
plt.ylabel("Billing Amount (in thousands)")
plt.xlabel("Age")
plt.figure(figsize = (10, 6))
plt.show()
```



<Figure size 1000x600 with 0 Axes>

Summary of the Joint plot: The Joint plot of Astham treatment in emergency care provides additional insights on the scatter plot and denotes that most of the treatment were given to people of ages between 55-65.Many people spent between 20-25k in emergency care for treating Asthma.

In [24]:
```
# Creating a Dataframe of Cancer patients that uses Medicare
```

```
cancer_df1=hosp_df[ (hosp_df["Medical Condition"].isin(["Cancer"]))   &
                   (hosp_df["Insurance Provider"].isin(["Medicare"])) ]
cancer_df1["Billing Amount"]=cancer_df1["Billing Amount"]/1000
# Creating a FacetGrid of Age vs Number of hospital stays based on the Gender
facetg = sns.FacetGrid(cancer_df1, hue = 'Gender', height = 7.5)
facetg.map(plt.scatter, 'Age', 'hospital_stay').add_legend()
plt.title("Hospital stays vs Age of Cancer patients on Medicare")
plt.ylabel("Hospital stay (in days)")
plt.xlabel("Age")
plt.figure(figsize=(8, 6))
plt.show()
```


Hospital stays vs Age of Cancer patients on Medicare
Hospital stay (in days)
0
5
10
15
20
25
30
20
30
40
50
60
70
80
Gender
Male
Female


```
<Figure size 800x600 with 0 Axes>
```

Summary of the FacetGrid: The plot indicates that fewer Male patients stayed in the hospital aged 30 and below compared to the female patients. Female patients stayed longer in the hospitals compared to the male, overall.

5.Create any additional visualizations that will help to answer the question(s) you want to answer.

5.1 Creating a Pie Chart

In [25]:
```
# Creating  a new dataframe for people of ages 65 and above
senior_df1=hosp_df[(hosp_df["Age"]>=65)]
# Grouping the data  based on the medical condition and getting the counts
senior_df1=senior_df1.groupby(["Medical Condition"]).count()["Name"].reset_index()
# Sorting the results based on the count
senior_df1=senior_df1.sort_values(by=["Name"], ascending=False)
```

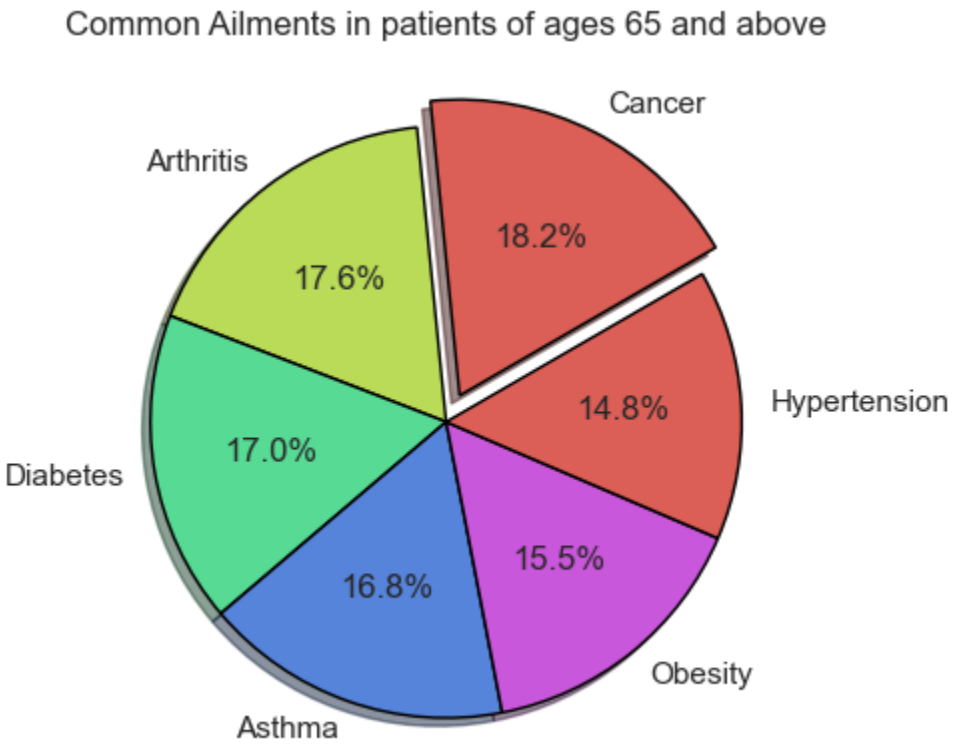
In [26]:
```
# Displaying  the data in the dataframe
senior_df1
```

Out[26]:

	Medical Condition	Name
2	Cancer	168
0	Arthritis	162
3	Diabetes	157
1	Asthma	155
5	Obesity	143
4	Hypertension	136

In [27]:
```
# Using explode to highlight the top category
explode=[0.1,0,0,0,0,0]
```

```
# Creating Pie chart with shadow enabled and displaying percent details.
plt.pie(senior_df1["Name"],labels=senior_df1["Medical Condition"],explode=explode,shadow=True, startangle=30,
        autopct="%1.1f%%",wedgeprops={'edgecolor':'black'},
       colors=sns.color_palette("hls", 5))
# Setting the plot title
plt.title("Common Ailments in patients of ages 65 and above")
plt.figure(figsize=(20, 15))
plt.show()
```




```
<Figure size 2000x1500 with 0 Axes>
```

Summary of the Pie plot: The pie plot indicates that Cancer was the most ailment in patients of ages 65 and above, closely followed by Arthiritis , Diabetes and Asthma.

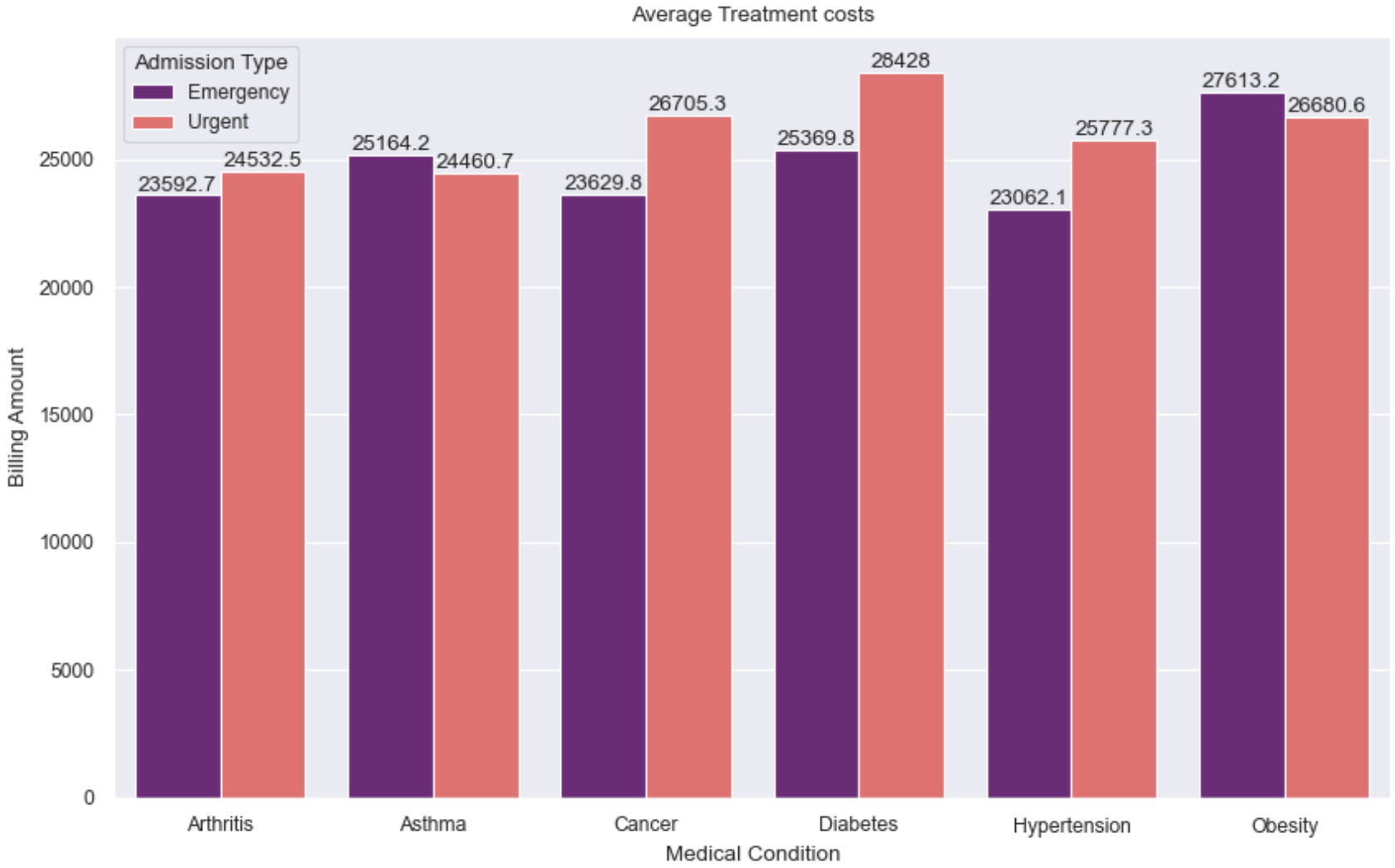
5.2 Creating a Bar plot

In [28]:
```
# Creating a dataframe to include the Urgent and Emergency admission types and Aetna and Cigna Providers
```

```
hosp_df4=hosp_df[ (hosp_df["Admission Type"].isin(["Urgent","Emergency"]))   &
                  (hosp_df["Insurance Provider"].isin(["Cigna","Aetna"])) ]
# Grouping the results based on the medical condition and the admission type and getting the average amount
hosp_df4=hosp_df4.groupby(["Medical Condition","Admission Type"]).mean()["Billing Amount"].reset_index()
```

In [29]:
```
# Setting the figure size and title of the barplot
sns.set_context('paper')
plt.figure(figsize=(10, 6))
plt.title("Average Treatment costs")
# creating the Barplot
ax_plot=sns.barplot(x = 'Medical Condition',
            y = 'Billing Amount',
            hue = 'Admission Type',
            data = hosp_df4,
           palette = 'magma')
# Adding Labels with values
ax_plot.bar_label(ax_plot.containers[0])
ax_plot.bar_label(ax_plot.containers[1])
```

Out[29]:
```
[Text(0, 0, '24532.5'),
 Text(0, 0, '24460.7'),
 Text(0, 0, '26705.3'),
 Text(0, 0, '28428'),
 Text(0, 0, '25777.3'),
 Text(0, 0, '26680.6')]
```



Summary of the Bar plot: The plot indicates that Urgent care was surprisingly more expensive than Emergency on average, in treating Arthritis,Cancer, Diabetes and HyperTension.

6.Summarize your results and make a conclusion. Explain how you arrived at this conclusion and how your visualizations support your conclusion

Based on the observations of the plots, below results can be concluded:

1. The Histograms and Boxplots indicates that the columns such as Age, Bill amount are not normally distributed and also don't have any outliers.

2. The Barplot of the Diabetes patient admissions indicated that the year 2021 had more Diabetes patient admissions compared to previous years, while 2020 had the lowest. The months April and May were the busiest in treating Diabetes.

3. The Lineplot of Asthma patients categorized by Gender indicated that most of the admissions were in April and May, then peaking again in December.

4. The scatter plot of Asthma treatment indicated that the costs can vary from a few thousands upto 50k in Emergency. Also most people spent between 20-25k in emergency.

5. The Facetgrid plot indicated that female cancer patients stayed longer in the hospital compared to Male on average.

6. Cancer was found to the most common disease in the people of ages 65 and above, closely followed by Arthritis, Diabetes and Asthma. These results was found using the PiePlot.

7. Surprisingly, the Urgent care turned out to be slightly expensive compared to Emergency care for treating Arthritis,Cancer, Diabetes and HyperTension.

These observations also helped to answer the four questions that were mentioned in the beginning of the project.

i. What is the common health condition in both male and female patients aged 65 and above?

Observations: Cancer was found to the most common disease in the people of ages 65 and above, closely followed by Arthritis, Diabetes and Asthma

ii. During which month of the year were more patients treated for Diabetes?

Observations: The year 2021 had more Diabetes patient admissions compared to previous years, while 2020 had the lowest. The months April and May were the busiest in treating Diabetes.

iii.What is the average amount billed for treating in Emergency vs Urgent Care?

Observations: The Urgent care turned out to be slightly expensive compared to Emergency care for treating Arthritis,Cancer, Diabetes and HyperTension.

iv. Is there any pattern in the Patient stays in hospitals for cancer treatment?

Observations: The female cancer patients stayed longer in the hospital compared to Male on average.