Diabetes Project

Project Description

The age, gender, body mass index (BMI), and blood pressure (BP) of certain individuals will be assessed for a study related to a diabetes project. We aim to explore relationships and potential interactions among these variables to gain valuable insights into the diabetes study.

Project Questions

Some of the questions we want to answer in this reseach are the following:

1. How does **age** affect **blood pressure**, and is there a particular age range where blood pressure tends to be highest or lowest?

2. Is there a **gender-based** difference in **BMI** across different age groups, and how does this gender-age interaction impact overall health?

3. What is the relationship between **BMI** and **blood pressure**, and are there specific BMI categories associated with elevated or reduced blood pressure?

4. "There is a significant gender-based difference in the average BMI among adults." This hypothesis is going to be tested using a two-sample t-test to compare the mean BMI of males and females. This study can help to determine whether gender is associated with differences in BMI and whether specific gender-based health interventions are necessary.

Project Procedures

Data will be loaded into a data frame, review for errors or inconsistencies, and clean it for analysis. Python is the language use for this study.

Tables and graphs would show some statistics and visuals for better understanding.

Project Libraries

Modules used in this study.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
%matplotlib inline
from IPython.display import display
```

Project Data

A dataset called diabetes.csv was download from a realiable source for the study.

In [2]:
```
# Read the data set and create a Data frame:
diabetes = r"C:\Users\rvrei\OneDrive\Desktop\Data Analyst\Datasets\Diabetes.csv"
df_raw = pd.read_csv(diabetes)
df_raw.head()
```

Out[2]:

	AGE SEX BMI BP S1 S2 S3 S4 S5 S6 Y
0	59 2 32.1 101 157 93.2 ...
1	48 1 21.6 87 183 103.2 ...
2	72 2 30.5 93 156 93.6 ...
3	24 1 25.3 84 198 131.4 ...
4	50 1 23 101 192 125.4 ...

In [3]:
```
# Evaluate the size (or dimensionality) of the data frame (rows and columns):
df_raw.shape
```

Out[3]:
```
(442, 1)
```

The dataframe created "df_raw" has 442 rows and only 1 column. It seems that we are dealing with a "single column dataframe" where all the data is in one column.

We will make some transformation to the dataframe to represent all the data among different columns as a table.

DataFrame Transformation

In [4]:
```
# List of the features names:
df_raw.columns.tolist()
```

Out[4]:
```
['AGE     SEX     BMI     BP      S1      S2      S3      S4      S5      S6
Y']
```

In [5]:
```
# Rename the single column list as "Combined_Column" name
df_raw = df_raw.rename(columns={'AGE     SEX     BMI     BP      S1      S2      S3
df_raw.head()
```

Out[5]:

	Combined_Column
0	59 2 32.1 101 157 93.2 ...
1	48 1 21.6 87 183 103.2 ...
2	72 2 30.5 93 156 93.6 ...
3	24 1 25.3 84 198 131.4 ...
4	50 1 23 101 192 125.4 ...

We split the single column "Combined_Column" into multiple ones, each one with its corresponding values. For that, we count the possible columns we would have from our dataframe based on 'space' delimeter.

In [6]:
```
# Count the number of columns (commas + 1)
delimiter = ' '
num_columns = df_raw['Combined_Column'].str.count(delimiter).max() + 1
print(f"Number of columns: {num_columns}")
```

```
Number of columns: 59
```

The count of possible columns shows 59 separations inside the single column dataframe.

In [7]:
```
# Split the 'Combined_Column' into 59 columns based on 'space' delimeter:
df_split = df_raw['Combined_Column'].str.split(delimiter, expand=True)

# Rename the columns as a range from column 1 to 59
df_split.columns = df_split.columns = [i for i in range(1,60)]

# Display the resulting Data Frame
pd.set_option('display.max_columns', None)   # To be able to see all columns
display(df_split.head())
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	2
0	59						2							32.1				101					1
1	48						1							21.6				87					
2	72						2							30.5				93					
3	24						1							25.3				84					
4	50						1							23						101			

We have now 59 columns in our dataframe. The data is spread over those columns, and we have to reorganize those values into a more meaningful columns.

First, we will clean some of the empty rows, and then we will consolidate the values into the first 11 columns: 'AGE', 'SEX', 'BMI', 'BP', 'S1', 'S2', 'S3', 'S4', 'S5', 'S6', 'Y'.

In [8]:
```
# Drop empty rows:
rows_to_drop = [108, 167, 199, 227, 302, 317, 321, 322, 380, 416, 426]
df1 = df_split.drop(rows_to_drop)
df1.head()
```

Out[8]:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	59						2							32.1				101				
1	48						1							21.6				87				
2	72						2							30.5				93				
3	24						1							25.3				84				
4	50						1							23						101		

In [9]:
```
# Replace the empty values in the data frame with NaN:
df1 = df1.replace({'': np.nan, None: np.nan})
display(df1.head())
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	59	NaN	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	32.1	NaN	NaN
1	48	NaN	NaN	NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	21.6	NaN	NaN
2	72	NaN	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	30.5	NaN	NaN
3	24	NaN	NaN	NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	25.3	NaN	NaN
4	50	NaN	NaN	NaN	NaN	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	23	NaN	NaN

Consolidate Values

Bring all the values to the important columns

In [10]:
```
# Create a new data frame after consolidation. Only keeping four columns: AGE, SEX,
new_data = {
    'AGE':df1[1],    # AGE new_column with the values of column 1
    'SEX':df1[7],    # SEX new_column with the values of column 7
    'BMI':df1[14],   # BMI new_column with the values of column 14
    'BP':df1[18]     # BP new_column with the values of column 18
}
```

```
consolidate_df = pd.DataFrame(new_data)
consolidate_df.head()
```

Out[10]:

	AGE	SEX	BMI	BP
0	59	2	32.1	101
1	48	1	21.6	87
2	72	2	30.5	93
3	24	1	25.3	84
4	50	1	23	NaN

In [11]:
```
# Checking missing values in the 'consolidate_df' data frame
consolidate_df.isnull().sum()
```

Out[11]:
```
AGE     0
SEX     0
BMI     0
BP     64
dtype: int64
```

There are 64 missing values in BP (Blood Pressure) column.

We substitute those missing values with the values from column 20.

In [12]:
```
# List of all non-NaN values of column 20
substitute_values = df1[20].dropna().tolist()
len(substitute_values)   # Check if the values from column 20 match the missing val
```

Out[12]:
```
64
```

In [13]:
```
# Create a mask for the 64 NaN values from column BP
nan_mask = consolidate_df['BP'].isna()

# Replace the 64 NaN values from 'BP' column with the values from the list of value
consolidate_df.loc[nan_mask, 'BP'] = substitute_values

# Display the modified Data Frame
consolidate_df.head()
```

Out[13]:

	AGE	SEX	BMI	BP
0	59	2	32.1	101
1	48	1	21.6	87
2	72	2	30.5	93
3	24	1	25.3	84
4	50	1	23	101

In [14]:
```
# Check again if there is any missing values in our 'consolidate_df' data frame
consolidate_df.isnull().sum()
```

Out[14]:
```
AGE     0
SEX     0
BMI     0
BP      0
dtype: int64
```

Our dataframe is free of NaN values. We will check data types and final clean ups.

In [15]:
```
# See the data type of each column in the data frame
consolidate_df.dtypes
```

Out[15]:
```
AGE     object
SEX     object
BMI     object
BP      object
dtype: object
```

In [16]:
```
# We change the data type of some columns to its nature
consolidate_df['AGE'] = consolidate_df['AGE'].astype(int)
consolidate_df['BMI'] = consolidate_df['BMI'].astype(float)
consolidate_df['BP'] = consolidate_df['BP'].astype(float)
```

In [17]:
```
print(f"Data type of 'AGE' column: {consolidate_df['AGE'].dtype}")
print(f"Data type of 'SEX' column: {consolidate_df['SEX'].dtype}")
print(f"Data type of 'BMI' column: {consolidate_df['BMI'].dtype}")
print(f"Data type of 'BP' column: {consolidate_df['BP'].dtype}")
```

```
Data type of 'AGE' column: int32
Data type of 'SEX' column: object
Data type of 'BMI' column: float64
Data type of 'BP' column: float64
```

Stratification

Replace number values for category names

In [18]:
```
# Stratification of the column 'SEX' with a new column 'GENDER', we replace numbers
# Stratification of the column 'AGE' with a new column 'AGE_strata', we create grou
consolidate_df['GENDER'] = consolidate_df['SEX'].replace({'1':'Male', '2':'Female'}
consolidate_df['AGE_strata'] = pd.cut(consolidate_df['AGE'], 3, labels=['19-39','40
consolidate_df.head()
```

Out[18]:

	AGE	SEX	BMI	BP	GENDER	AGE_strata
0	59	2	32.1	101.0	Female	40-59
1	48	1	21.6	87.0	Male	40-59
2	72	2	30.5	93.0	Female	60+
3	24	1	25.3	84.0	Male	19-39
4	50	1	23.0	101.0	Male	40-59

In [19]:
```
# Save the clean data frame to a csv file
consolidate_df.to_csv('clean_diabetes.csv', columns=['AGE','SEX','BMI','BP','GENDER
```

In [20]:
```
# Clean data frame with the only the columns needed
df = consolidate_df.loc[:, ['AGE','GENDER','BMI','BP','AGE_strata']]
df.head()
```

Out[20]:

	AGE	GENDER	BMI	BP	AGE_strata
0	59	Female	32.1	101.0	40-59
1	48	Male	21.6	87.0	40-59
2	72	Female	30.5	93.0	60+
3	24	Male	25.3	84.0	19-39
4	50	Male	23.0	101.0	40-59

Clean dataframe: df
Columns:

> AGE = Ages of the sample people

GENDER = Male, Female
BMI = Body Mass Index
BP = Average Blood Pressure
AGE_strata = Age groups: (19-39, 40-59, 60+)

Descriptive Analysis

In [21]:
```
# Summary Statistics of the data set
df.describe().transpose().round(2)
```

Out[21]:

	count	mean	std	min	25%	50%	75%	max
AGE	431.0	48.28	13.11	19.0	38.0	50.0	59.00	79.0
BMI	431.0	26.33	4.41	18.0	23.1	25.7	29.20	42.2
BP	431.0	94.25	13.75	62.0	84.0	93.0	103.84	133.0

We have a table with the summary statistic for the variables: AGE, BMI, and BP in general.

From this statistics we can extract some information per variable as:

> AGE:
>
> > The ages consider for our studies range from 19 years old to 79 years old.
>
> The median age would be 50 years old with the average being 48.28 years old for the individuals in our study.

> BMI:
>
> > The range of BMI goes from 18 to 42.2.
> > The median is 25.7 and the average 26.33.

> BP:
>
> > The range of BP goes from 133 to 62.
> > The median is 93 and the average 94.25.

In [22]:
```
# Summary Statistics per Gender
df.groupby('GENDER').describe().stack().transpose().round(2)
```

Out[22]:

GENDER									**Female**				
	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**	**count**	**mean**	**std**	**min**	**2**
AGE	203.0	50.80	12.76	20.0	41.0	53.0	60.0	79.0	228.0	46.04	13.04	19.0	3(
BMI	203.0	26.78	4.21	18.0	24.0	25.9	28.9	42.2	228.0	25.93	4.56	18.5	2(
BP	203.0	97.99	12.64	70.0	89.0	97.0	109.0	126.0	228.0	90.92	13.86	62.0	8(

Splitting the summary statistics per gender to compare between male and female.
In general, the first observation is the higher values for females than males for BMI and BP.
We will have to see later if those differences are statistically significant to make a proper evaluation of those values.

In [23]:
```
# Summary statistics per Gender and Age's Group
df.groupby(['AGE_strata','GENDER']).describe().round(2)
```

Out[23]:

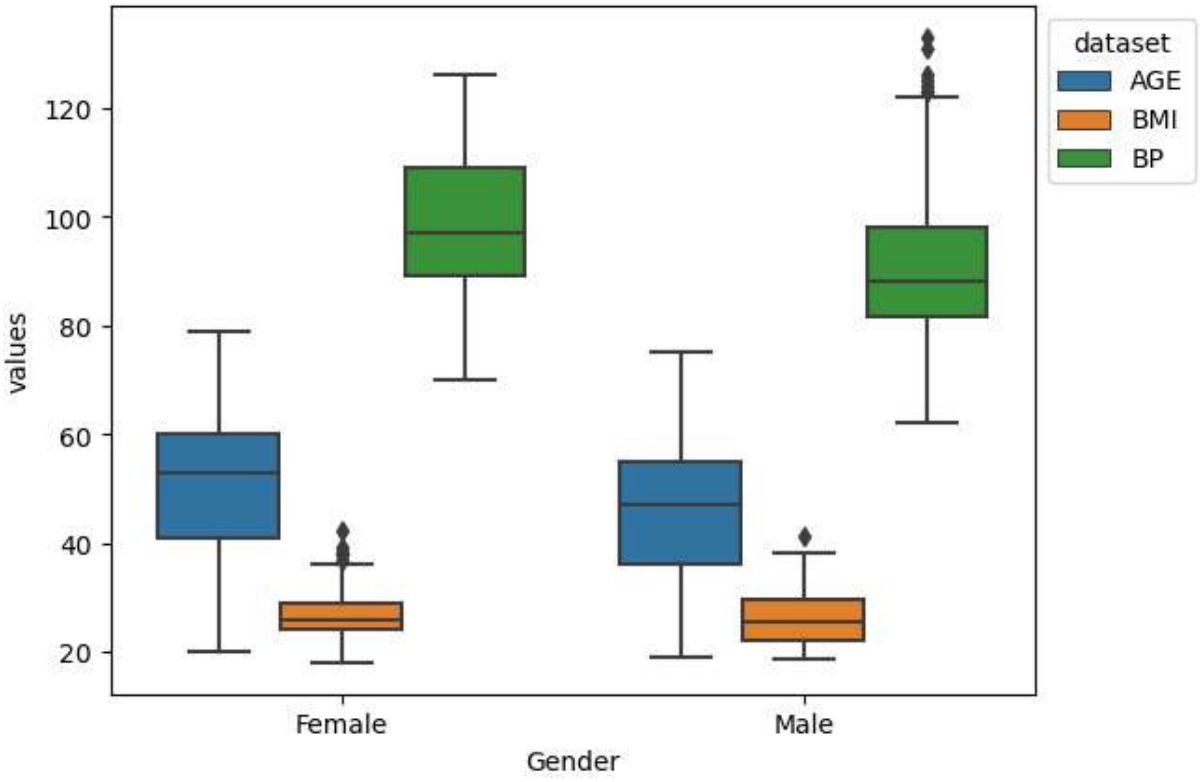
		AGE										
		count	mean	std	min	25%	50%	75%	max	count	mean	std
AGE_strata	GENDER											
19-39	Female	46.0	32.11	5.03	20.0	28.25	33.0	36.0	39.0	46.0	25.74	4.92
	Male	71.0	30.52	5.94	19.0	25.50	32.0	35.5	39.0	71.0	24.90	4.96
40-59	Female	98.0	51.15	5.55	40.0	47.00	51.5	56.0	59.0	98.0	27.01	4.09
	Male	117.0	48.97	5.07	40.0	44.00	49.0	53.0	58.0	117.0	26.25	4.41
60+	Female	59.0	64.78	4.46	60.0	61.00	64.0	67.0	79.0	59.0	27.20	3.73
	Male	40.0	65.00	4.19	60.0	61.00	64.5	68.0	75.0	40.0	26.81	3.98

Finally, we split our data into gender and group of ages to see any changes in BMI and BP in them.

Visualization

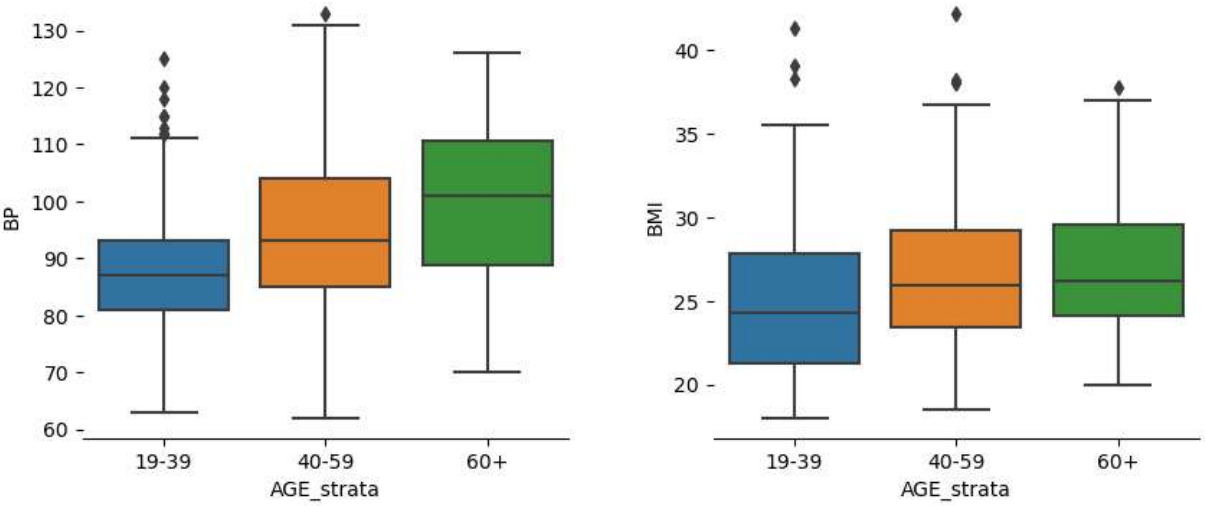
In [24]:
```
# Boxplot of the variables by Gender
df_ = df.loc[:, df.columns != 'AGE_strata']   # We drop the age's group column for
data = df_.melt(['GENDER'], var_name='dataset', value_name='values')

sns.boxplot(data=data, x='GENDER', y='values', hue='dataset')
plt.legend(title='dataset', loc='upper left', bbox_to_anchor=(1, 1))
plt.xlabel('Gender');
```



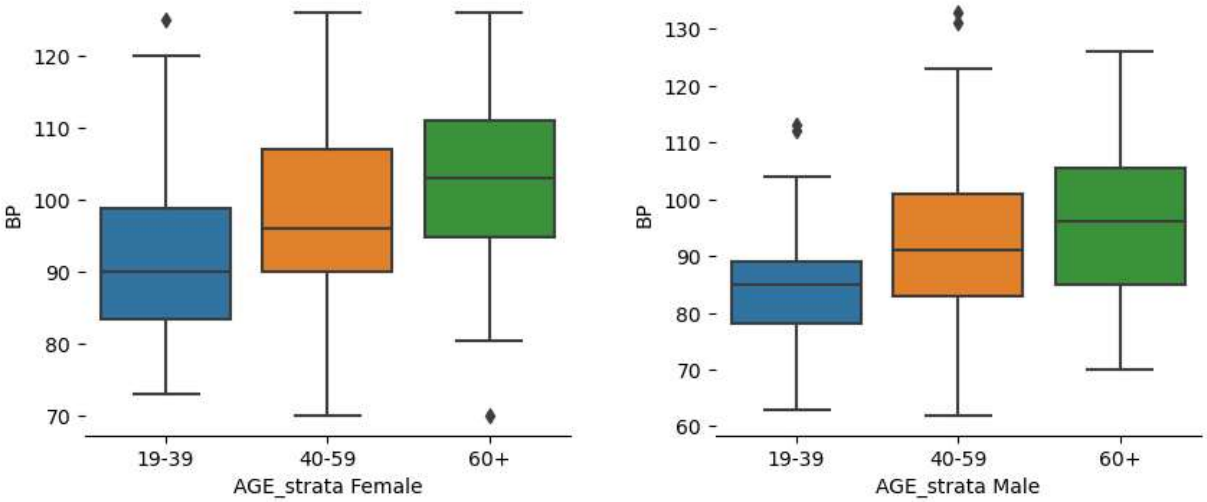
We can see some outliers present for 'BMI' and, in males for 'BP'

In [25]:
```
# Boxplot of Blood Pressure and Boby Mass Index by Age group
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(10,4))
sns.despine(left=True)
sns.boxplot(x='AGE_strata', y='BP', data=df, ax=ax1)
sns.boxplot(x='AGE_strata', y='BMI', data=df, ax=ax2)
plt.subplots_adjust(wspace=0.3)
plt.show();
```



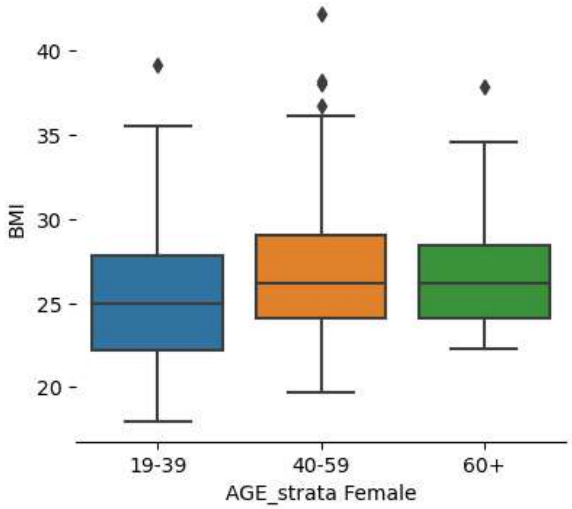
One clear distinction can be seen from Blood Pressure and Body Mass Index by age group, Age is an important factor for the values to increase. We observed that the median is getting higher the older someone gets.

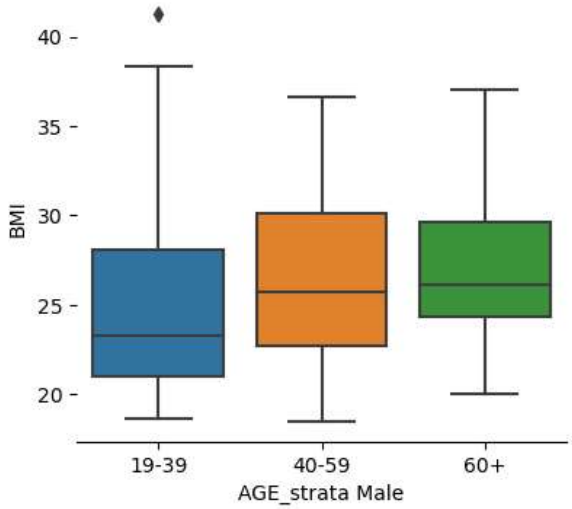
In [26]:
```
# Boxplot of Blood Pressure and Age group separated by gender
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(10,4))
sns.despine(left=True)
sns.boxplot(x=df.loc[df['GENDER']=='Female','AGE_strata'], y='BP', data=df, ax=ax1)
ax1.set(xlabel='AGE_strata Female')
sns.boxplot(x=df.loc[df['GENDER']=='Male','AGE_strata'], y='BP', data=df, ax=ax2)
ax2.set(xlabel='AGE_strata Male')
plt.subplots_adjust(wspace=0.3)
plt.show();
```



As we observed in the previous plot, the age median increases clearly with age in both female and male for our sample individuals. Also, we notice a few outliers in our dataset that we will handle later on.

In [27]:
```
# Boxplot of Body Mass Index and Age group separated by gender
fig, (ax3, ax4) = plt.subplots(nrows=1, ncols=2, figsize=(10,4))
sns.despine(left=True)
sns.boxplot(x=df.loc[df['GENDER']=='Female','AGE_strata'], y='BMI', data=df, ax=ax3
ax3.set(xlabel='AGE_strata Female')
sns.boxplot(x=df.loc[df['GENDER']=='Male','AGE_strata'], y='BMI', data=df, ax=ax4)
ax4.set(xlabel='AGE_strata Male')
plt.subplots_adjust(wspace=0.3)
plt.show();
```





For Body Mass Index, it seems that there are a break in the age group between 19-39 years old and 40+. The median BMI has a higher increase for males than females but stays stable for ages above 40. Another clear distinction between males and females is the higher range of BMI values for males than females. It seems we have higher variety of values for males, however the median are smaller for males than females.

We will study those outliers more deeply to evaluate and decide what to do with them.

We will have outliers for Blood Pressure if its value are out the following range:

> (Q1_bp - 1.5*IQR_bp) < *Blood Pressure values* < (Q3_bp + 1.5IQR_bp)

In [28]:
```
#Find the 1st and 3rd quartile:
Q1_bp = df.BP.quantile(0.25)
Q3_bp = df.BP.quantile(0.75)
IQR_bp = Q3_bp - Q1_bp
no_outliers_bp = df.BP[(Q1_bp - 1.5*IQR_bp < df.BP) &  (df.BP < Q3_bp + 1.5*IQR_bp)
outliers_bp = df.BP[(Q1_bp - 1.5*IQR_bp >= df.BP) | (df.BP >= Q3_bp + 1.5*IQR_bp)]
print(outliers_bp)
```

```
Series([], Name: BP, dtype: float64)
```

It seems that we don't have any outliers for the variable Blood Pressure.

In [29]:
```
print(f'BP outlier <= {(Q1_bp - 1.5*IQR_bp).round(2)}')
print(f'BP outlier >= {(Q3_bp + 1.5*IQR_bp).round(2)}\n')
print(f"BP_max = {df['BP'].max()}\nBP_min = {df['BP'].min()}")
```

```
BP outlier <= 54.25
BP outlier >= 133.59

BP_max = 133.0
BP_min = 62.0
```

In [30]:
```
# Outliers for Boby Mass Index
Q1_bmi = df.BMI.quantile(0.25)
Q3_bmi = df.BMI.quantile(0.75)
```

```
IQR_bmi = Q3_bmi - Q1_bmi
no_outliers_bmi = df.BMI[(Q1_bmi - 1.5*IQR_bmi < df.BMI) &  (df.BMI < Q3_bmi + 1.5*
outliers_bmi = df.BMI[(Q1_bmi - 1.5*IQR_bmi >= df.BMI) |  (df.BMI >= Q3_bmi + 1.5*I
print(f'These are the outliers for BMI:\n{outliers_bmi}')
```

```
These are the outliers for BMI:
256    41.3
366    39.1
367    42.2
Name: BMI, dtype: float64
```

We have three outliers for the variable BMI. Because there are a small number we can drop it from our calculations.

In [31]:
```
print(f'BMI outlier <= {(Q1_bmi - 1.5*IQR_bmi).round(2)}')
print(f'BMI outlier >= {(Q3_bmi + 1.5*IQR_bmi).round(2)}\n')
print(f"BMI_max = {df['BMI'].max()}\nBMI_min = {df['BMI'].min()}")
```

```
BMI outlier <= 13.95
BMI outlier >= 38.35

BMI_max = 42.2
BMI_min = 18.0
```

In [32]:
```
# We remove the outliers from our data frame for the following calculations.
rows_to_drop = [256, 366, 367]
df_no_out = df.drop(rows_to_drop)
df_no_out.head()
```

Out[32]:

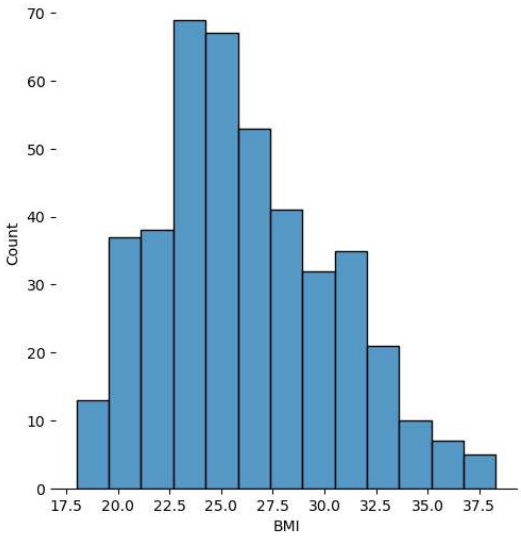
	AGE	GENDER	BMI	BP	AGE_strata
0	59	Female	32.1	101.0	40-59
1	48	Male	21.6	87.0	40-59
2	72	Female	30.5	93.0	60+
3	24	Male	25.3	84.0	19-39
4	50	Male	23.0	101.0	40-59

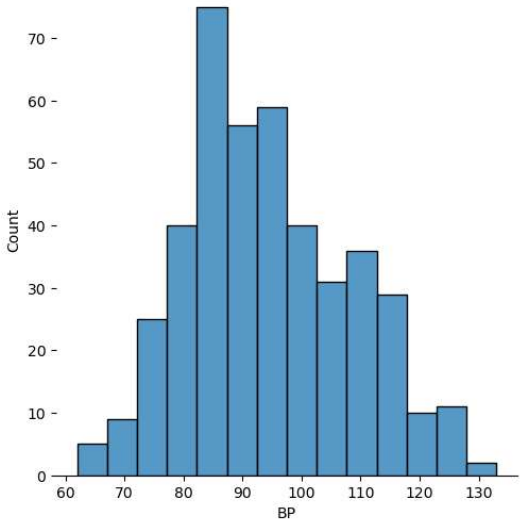
In [33]:
```
# Histogram of BMI and BP
df_bmi = df_no_out.loc[:, 'BMI']
df_bp = df_no_out.loc[:, 'BP']

fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(14,6))
sns.despine(left=True)
sns.histplot(df_bmi, ax=ax1)
ax1.set(xlabel='BMI')

sns.histplot(df_bp, ax=ax2)
ax2.set(xlabel='BP')

plt.subplots_adjust(wspace=0.5)
plt.show();
```





Body Mass Index (BMI):

- The distribution is unimodal and a little skewed to the right.
- It is centered at about 25.65 with most of the data between 18 and 38.3.
- The range is roughly 20.3, and the outliers are in the higher end.

Blood Pressure (BP):

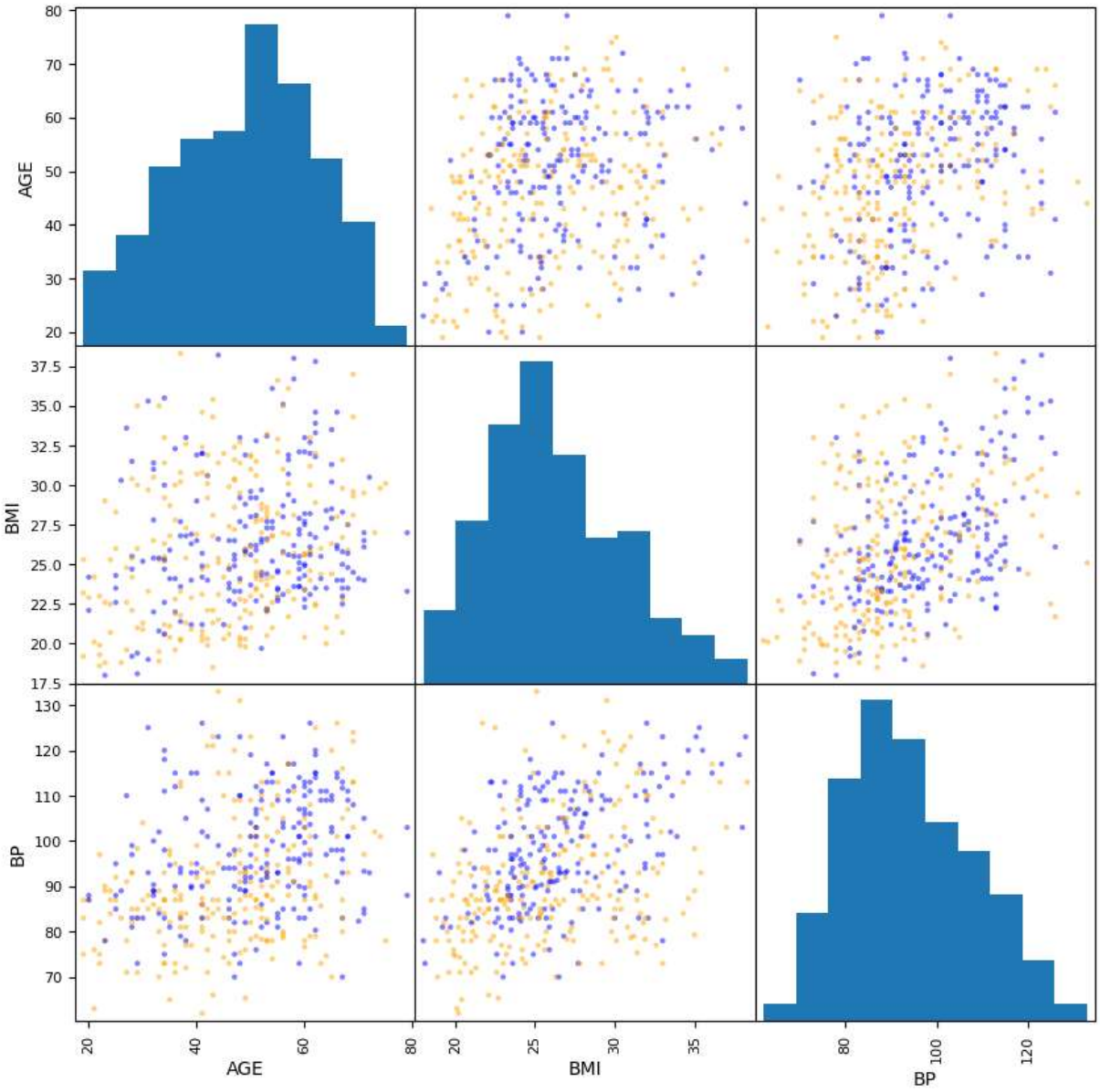
- The distribution is unimodal and a little skewed to the right.
- It is centered at about 93 with most of the data between 62 and 133.
- The range is roughly 71, and no outliers have been identified.

In [34]:
```
# Summary statistics for Body Mass Index and Blood Pressure:
df_no_out.loc[:, ['BMI','BP']].describe().transpose()
```

Out[34]:

	count	mean	std	min	25%	50%	75%	max
BMI	428.0	26.225467	4.255398	18.0	23.1	25.65	29.05	38.3
BP	428.0	94.278762	13.775211	62.0	84.0	93.00	104.00	133.0

In [35]:
```
# Scatter-Matrix of all our variables:
df_no_out['GENDERx'] = df_no_out['GENDER'].replace({'Female':'b', 'Male':'orange'})
from pandas.plotting import scatter_matrix
scatter_matrix(df_no_out, figsize=(10,10), c=df_no_out['GENDERx']);
```



In [36]:
```
# Correlation Coefficients between the variables:
corr= df_no_out.corr()
corr
```

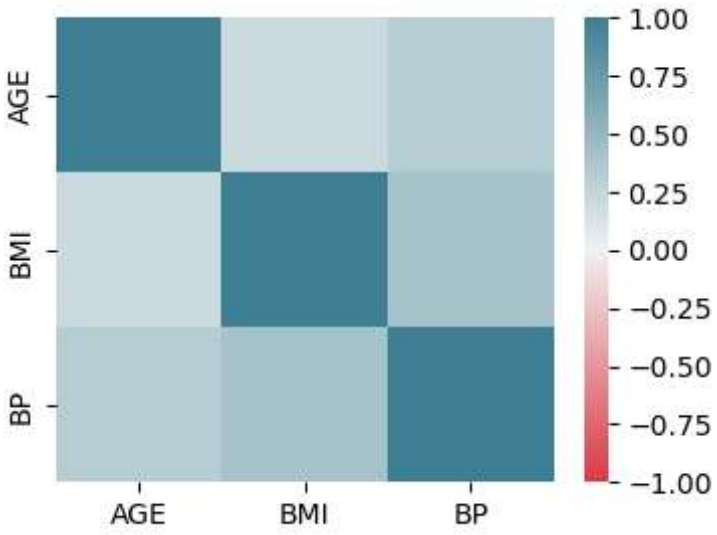
Out[36]:

	AGE	BMI	BP
AGE	1.000000	0.208811	0.324489
BMI	0.208811	1.000000	0.415464
BP	0.324489	0.415464	1.000000

We have small to medium correlation among our variables to draw any significant conclousion about them.

In [37]:
```
# Heatmap for correlation and scatter-matrix
cmap = sns.diverging_palette(10, 220, as_cmap=True)
```

```
plt.figure(figsize=(4, 3))
sns.heatmap(corr, vmin=-1.0, vmax=1.0, cmap=cmap);
```



Question 1

How does **age** affect **blood pressure**, and is there a particular age range where blood pressure tends to be highest or lowest?

- We will attempt to identify any significant implications between 'Age' and 'Blood Pressure' in our study, even though the correlation coefficient indicates a minor relationship between these two variables.
- We will assess this relationship separately for different genders, examining males and females to identify any interesting differences between them.
- Additionally, we will categorize the 'Age' variable into age groups to observe any patterns or trends between age and blood pressure that could be useful for our study.

We obtained the following data on blood pressure from the CDC:

Normal Blood Pressure is less than 120/80 mm Hg

- For men 124/72 mm Hg

  > 18-39 years: 119/70 mm Hg

40-59 years: 124/77 mm Hg
60+ years: 130/69 mm Hg

- For women 121/70 mm Hg

  > 18-39 years: 110/68 mm Hg

40-59 years: 122/74 mm Hg
60+ years: 139/68 mm Hg

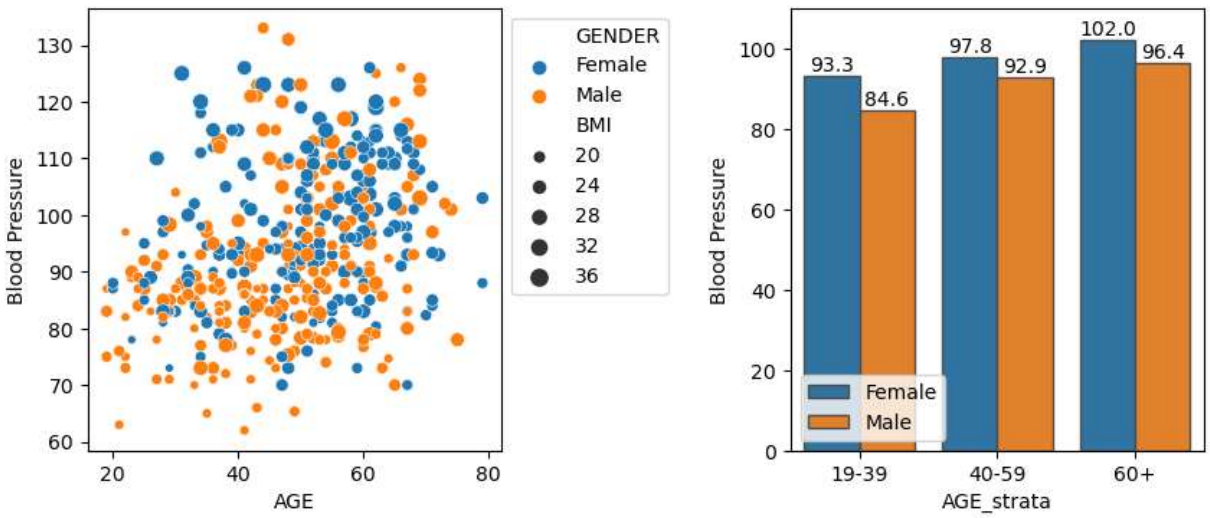
Hypertension is defined as having a systolic pressure of 130 mm Hg or higher, or a diastolic pressure of 80 mm Hg or higher.

In [38]:
```
# Create 2 plots side by side of Blood Pressure vs. Age:
fig, (ax1,ax2) = plt.subplots(nrows=1, ncols=2, figsize=(10, 4))

# 1st Plot
g1 = sns.scatterplot(data=df_no_out, x='AGE', y='BP', hue='GENDER', size='BMI', ax=
ax1.set(ylabel='Blood Pressure')
g1.legend(loc='upper left',bbox_to_anchor=(1,1))

# 2nd Plot
g2 = sns.barplot(x='AGE_strata', y='BP', data=df_no_out, errwidth=0, hue='GENDER',
# add the annotation
for i in ax2.containers:
    ax2.bar_label(i,fmt='%.1f', label_type='edge')
ax2.set(ylabel='Blood Pressure')
g2.legend(loc=3)

plt.subplots_adjust(wspace=0.7)   # Adjust the space between both plots
plt.show();
```



Results:

- The relationship between Blood Pressure and Age for males and females are scattered across the whole range, as the correlation coefficient showed us before.
- For the Age groups the Blood Pressure on average shows a slowly increment with age for males and females, indicating that males tend to have lower Blood Pressure than females on average for the individuals in our data.

In [39]:
```
# Summary statistics of Blood Pressure by Gender
df_no_out.groupby('GENDER').describe()['BP'].round(2)
```

Out[39]:

GENDER	count	mean	std	min	25%	50%	75%	max
Female	201.0	98.02	12.69	70.0	89.0	97.0	109.00	126.0
Male	227.0	90.96	13.87	62.0	82.0	88.0	98.16	133.0

We observe a significant variation in average blood pressure between males and females. To ensure the statistical significance of these averages and consider them in our analysis, we will conduct a 'Two-Sample t-test for the mean' of blood pressure in both genders.

- Null Hypothesis: bp_mean_male = bp_mean_female
- Alternative Hypothesis: bp_mean_male != bp_mean_female

If the p_value is less than the significance level alpha of 0.05, then we can reject the null hypothesis.

For this t-test statistic we assumed that one sample is independent of the other, the data is approximately normally distributed, the two samples have approximately the same variance, and the data in both samples was obtained using a random sampling method.

In [40]:
```
# We create the two groups of BP, males and females:
group1_male = df_no_out[df_no_out['GENDER']=='Male']['BP']
group2_female = df_no_out[df_no_out['GENDER']=='Female']['BP']
```

In [41]:
```
# Print the variance values for each BP group:
print(f"Variance_male = {round(np.var(group1_male),2)}\nVariance_female = {round(np
```

```
Variance_male = 191.64
Variance_female = 160.24
```

In [42]:
```
# Print the Ratio of variances of both BP groups:
print(f"Ratio of variances = {np.var(group1_male)/np.var(group2_female)}")
```

```
Ratio of variances = 1.1959450183565221
```

Because the ratio of the larger sample variance to the smaller sample variance is 1.1959, which it is less than 4, this means that we can assume that the population variances are equal.

We check if the variances can really be considered as equal with the Bartlett's Test.

- Null Hypothesis variance_male = variance_female
- Alternative Hypothesis: both variances are not equal

In [44]:
```
import scipy.stats as stats
stats.bartlett(group1_male, group2_female)
```

Out[44]:
```
BartlettResult(statistic=1.6755074799009575, pvalue=0.19552247386308125)
```

Since the p_value=0.1955 is greater than alpha=0.05, we fail to reject the null hypothesis that says that both variances are equal. We don't have sufficient evidence to say that the two groups have different variances.

In [46]:
```
# Two samples t-test of the mean blood pressure for males and females:
import scipy.stats as stats
stats.ttest_ind(a=group1_male, b=group2_female, equal_var=True)
```

Out[46]:
```
TtestResult(statistic=-5.467803993239184, pvalue=7.77739673943009e-08, df=426.0)
```

The p_value is 7.7774e-08, which is smaller than our alpha of 0.05 (significance level). Therefore, we reject the null hypothesis that states bp_mean_male = bp_mean_female. We have sufficient evidence to say that bp_mean_male and bp_mean_female are different.

In [47]:
```
# Range in Blood Pressure data by gender
df_no_out.groupby('GENDER')['BP'].max() - df_no_out.groupby('GENDER')['BP'].min()
```

Out[47]:
```
GENDER
Female    56.0
Male      71.0
Name: BP, dtype: float64
```

Observations of Blood Pressure vs. Gender in our sample dataset:

- We have 26 more males samples than females.
- The range per gender is:
  - Males = 71
  - Females = 56
  - This reflects that males values of Blood Pressure are more spread out than the females ones, confirmed by the higher standard deviation for males than females.
- Both male and female distributions are slightly skewed to the right, where their mean values are a little higher than their median/center values.
- On average females have 7.2% higher Blood Pressure than males in our sample dataset.
- The data indicate that males reach a maximum highest of blood pressure in average than females. That confirm the higher range for males than females. It looks like the males group has some individuals that have high values in blood pressure, because for the 75% of the male people, they are still with lower values than the females ones.

In [48]:
```
# Summary Statistics between Blood Pressure and Age Groups:
df_no_out.groupby(['AGE_strata']).describe()['BP'].round(2)
```

Out[48]:

AGE_strata	count	mean	std	min	25%	50%	75%	max
19-39	115.0	87.99	11.65	63.0	81.50	87.0	93.0	125.0
40-59	214.0	95.12	13.65	62.0	85.00	93.0	104.0	133.0
60+	99.0	99.77	13.62	70.0	88.84	101.0	110.5	126.0

Observation of Blood Pressure vs. Age Groups:

- It is observed how on average the Blood Pressure increase with each age group clearly in our dataset.
- The age group of 40 to 59 years old shows the higher maximum of Bloop Pressure values. Investigating a little more about this specific age group we found that its range is the biggest of the two other age groups, with a value of 71, showing that this age group has the most spread values over a higher range than the other groups.
- For the maximum Blood Pressure values, the youngest and the oldest group practically coincide. Conversely, for the minimum values, it is the youngest and middle age groups that align. This suggests that individuals in the age group of 60+ may noy reach the minimum blood pressure values seen in younger individuals. However, for the maximum values, stability is observed, particularly when compared to the youngest group.

In [49]:
```
# Summary Statistics between Blood Pressure and Age groups separated by Gender:
df_no_out.groupby(['AGE_strata','GENDER']).describe()['BP'].round(2)
```

Out[49]:

AGE_strata	GENDER	count	mean	std	min	25%	50%	75%	max
19-39	**Female**	45.0	93.26	12.77	73.0	83.00	90.0	99.0	125.0
	Male	70.0	84.60	9.52	63.0	78.00	85.0	89.0	113.0
40-59	**Female**	97.0	97.80	12.58	70.0	90.00	96.0	107.0	126.0
	Male	117.0	92.89	14.15	62.0	83.00	91.0	101.0	133.0
60+	**Female**	59.0	102.03	11.65	70.0	94.66	103.0	111.0	126.0
	Male	40.0	96.44	15.66	70.0	85.00	96.0	105.5	126.0

We observe distinctions between males and females in our dataset:

- In every age group, females have higher blood pressure than males. The most notable difference occurs in the age group of 19-39 years old, with a margin of 8.66 points.

What is the age range group where Blood Pressure tends to be the highest or lowest?

- According to our dataset, the highest blood pressure is observed in the age group between 40 and 59 years old. Additionally, we note that the youngest and the oldest age groups share almost the same maximum blood pressure.

- The lowest Blood Pressure is observed in the age group between 40 and 59 years old, with a similar pattern for the group between 19 and 39 years old.

- If we analyze the age groups separately for males and females, we observe the following:

For males, the highest blood pressure is observed in the age group between 40 and 59 years old, while the lowest blood pressure is also found in males in the same age group.

- Another interesting finding is the mean value for each age group. We observed that, on average, the older group has a higher blood pressure, with the youngest group having the lowest levels. This indicates that, on average, as we age, there is an increased likelihood of having higher blood pressure. This pattern is consistent when we analyze the age groups based on gender, with the youngest group having the lowest average blood pressure and the older group having the highest. Additionally, we note that females, in general, have a higher average blood pressure than males.

In [50]:
```
# Difference between males and females average on blood pressure per age groups
dif_19 = df_no_out[(df_no_out['AGE_strata']=='19-39') & (df_no_out['GENDER']=='Fema
dif_40 = df_no_out[(df_no_out['AGE_strata']=='40-59') & (df_no_out['GENDER']=='Fema
dif_60 = df_no_out[(df_no_out['AGE_strata']=='60+') & (df_no_out['GENDER']=='Female

print(f'19-39 age group Females/Males BP diff on average = {dif_19.round(2)}\n')
print(f'40-59 age group Females/Males BP diff on average = {dif_40.round(2)}\n')
print(f'60+ age group Females/Males BP diff on average = {dif_60.round(2)}')
```

```
19-39 age group Females/Males BP diff on average = 8.66

40-59 age group Females/Males BP diff on average = 4.91

60+ age group Females/Males BP diff on average = 5.59
```

```
C:\Users\rvrei\AppData\Local\Temp\ipykernel_20632\556349763.py:2: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  dif_19 = df_no_out[(df_no_out['AGE_strata']=='19-39') & (df_no_out['GENDER']=='Fem
ale')].mean()['BP'].round(2) - df_no_out[(df_no_out['AGE_strata']=='19-39') & (df_no
_out['GENDER']=='Male')].mean()['BP'].round(2)
C:\Users\rvrei\AppData\Local\Temp\ipykernel_20632\556349763.py:3: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  dif_40 = df_no_out[(df_no_out['AGE_strata']=='40-59') & (df_no_out['GENDER']=='Fem
ale')].mean()['BP'].round(2) - df_no_out[(df_no_out['AGE_strata']=='40-59') & (df_no
_out['GENDER']=='Male')].mean()['BP'].round(2)
C:\Users\rvrei\AppData\Local\Temp\ipykernel_20632\556349763.py:4: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  dif_60 = df_no_out[(df_no_out['AGE_strata']=='60+') & (df_no_out['GENDER']=='Femal
e')].mean()['BP'].round(2) - df_no_out[(df_no_out['AGE_strata']=='60+') & (df_no_out
['GENDER']=='Male')].mean()['BP'].round(2)
```

We observe that the difference in blood pressure among age groups and genders is significantly higher in the youngest ages. For the other age groups, the differences are relatively close.

Question 2

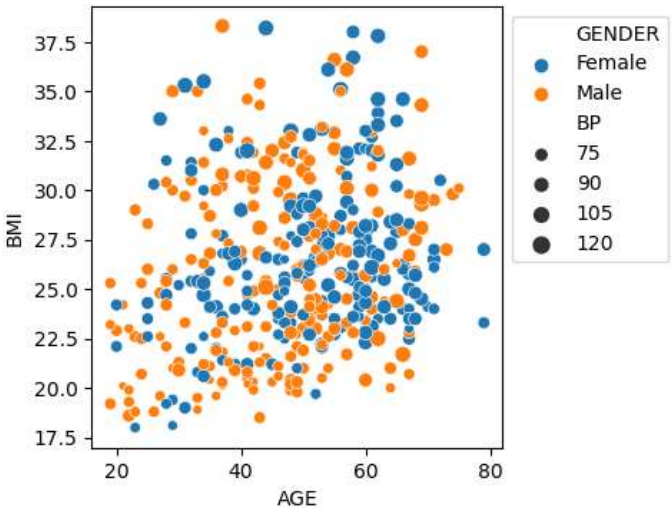
Is there a **gender-based** difference in **BMI** across different age groups, and how does this gender-age interaction impact overall health?

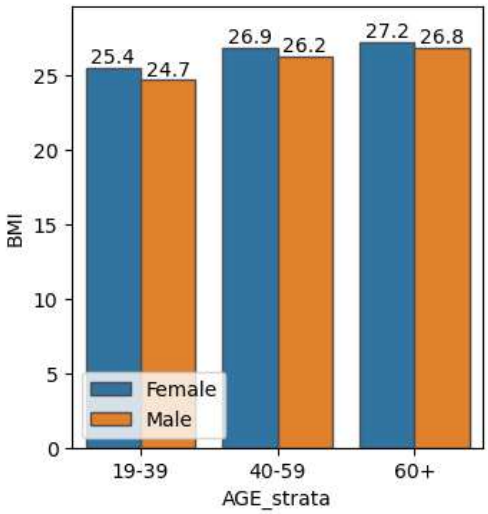
In [51]:
```
# Create 2 plots side by side of BMI vs AGE and, BMI vs Age groups:
fig, (ax3,ax4) = plt.subplots(nrows=1, ncols=2, figsize=(10, 4))

# 1st Plot
g3 = sns.scatterplot(data=df_no_out, x='AGE', y='BMI', hue='GENDER', size='BP', ax=
ax3.set(ylabel='BMI')
g3.legend(loc='upper left',bbox_to_anchor=(1,1))

# 2nd Plot
g4 = sns.barplot(x='AGE_strata', y='BMI', data=df_no_out, errwidth=0, hue='GENDER',
# add the annotation
for i in ax4.containers:
    ax4.bar_label(i,fmt='%.1f', label_type='edge')
ax4.set(ylabel='BMI')
g4.legend(loc=3)

plt.subplots_adjust(wspace=0.7)   # Adjust the space between both plots
plt.show();
```





- Males tend to have lower BMI than females in each age group within our dataset, mirroring the same trend observed for blood pressure vs. age.
- Additionally, there is a tendency for BMI to increase with age; on average, BMI values rise as individuals get older.
- The correlation coefficient suggests a poor correlation between BMI and age, with values scattered across the place.

In [52]:
```
# BMI standard according to CDC
bmi_standard = pd.DataFrame({'BMI':['below 18.5','18.5 - 24.9','25.0 - 29.9','30.0
                                    'Weight Status':['Underweight','Healthy weight','Overw
bmi_standard
```

Out[52]:

	BMI	Weight Status
0	below 18.5	Underweight
1	18.5 - 24.9	Healthy weight
2	25.0 - 29.9	Overweight
3	30.0 and above	Obesity

In [53]:
```
# Summary Statistics of BMI per Gender:
df_no_out.groupby('GENDER').describe()['BMI'].round(1)
```

Out[53]:

GENDER	count	mean	std	min	25%	50%	75%	max
Female	201.0	26.6	4.0	18.0	24.0	25.9	28.7	38.2
Male	227.0	25.9	4.4	18.5	22.1	25.4	29.4	38.3

- Once again, we observe a tendency for females to have a higher BMI than males on average.

- Interestingly, the maximum and minimum values of BMI are similar for both genders.

In [54]:
```
# We introduce a new column in our dataframe, BMI_strata, that categorizes BMI base
df_no_out['BMI_strata'] = pd.cut(x=df_no_out['BMI'], bins=[1,18.4,24.9, 28.9,np.inf
                                 labels=['Underweight', 'Healthy Weight','Overweigh
df_no_out.head()
```

Out[54]:

	AGE	GENDER	BMI	BP	AGE_strata	GENDERx	BMI_strata
0	59	Female	32.1	101.0	40-59	b	Obesity
1	48	Male	21.6	87.0	40-59	orange	Healthy Weight
2	72	Female	30.5	93.0	60+	b	Obesity
3	24	Male	25.3	84.0	19-39	orange	Overweight
4	50	Male	23.0	101.0	40-59	orange	Healthy Weight

In [55]:
```
# Summary statistics of BMI per gender and type of BMI:
df_no_out.groupby(['GENDER','BMI_strata']).describe()['BMI'].round(1)
```

Out[55]:

		count	mean	std	min	25%	50%	75%	max
GENDER	**BMI_strata**								
Female	**Underweight**	2.0	18.0	0.1	18.0	18.0	18.0	18.1	18.1
	Healthy Weight	76.0	23.2	1.4	19.0	22.6	23.5	24.1	24.9
	Overweight	74.0	26.6	1.0	25.0	25.7	26.5	27.5	28.8
	Obesity	49.0	32.4	2.4	29.0	30.6	32.0	33.5	38.2
Male	**Healthy Weight**	106.0	21.9	1.7	18.5	20.5	22.0	23.3	24.9
	Overweight	60.0	26.8	1.1	25.1	25.7	26.8	27.8	28.9
	Obesity	61.0	31.8	2.2	29.0	30.1	31.4	32.7	38.3

When comparing our dataset with the standard BMI from the CDC, we observe that only males in the age group of 19-39 years old fall within the Healthy Weight segment. In all other age groups, whether males or females, the average BMI is considered overweight. Notably, only females in the age group of 19-39 years old have two observations falling below the underweight category in our dataset.

In [56]:
```
# Summary statistics of BMI clasified by gender, age groups, and BMI weights:
df_no_out.groupby(['GENDER','AGE_strata','BMI_strata']).describe()['BMI'].round(1)
```

Out[56]:

GENDER	AGE_strata	BMI_strata	count	mean	std	min	25%	50%	75%	max
Female	**19-39**	**Underweight**	2.0	18.0	0.1	18.0	18.0	18.0	18.1	18.1
		Healthy Weight	21.0	22.4	1.8	19.0	21.2	22.6	24.1	24.7
		Overweight	12.0	26.3	0.9	25.2	25.4	26.1	26.8	27.8
		Obesity	10.0	32.4	1.9	30.0	31.1	31.9	33.4	35.5
	40-59	**Healthy Weight**	35.0	23.3	1.2	19.7	22.7	23.6	24.2	24.9
		Overweight	37.0	26.6	1.0	25.1	25.8	26.5	27.4	28.8
		Obesity	25.0	32.1	2.8	29.0	29.6	31.9	33.0	38.2
	60+	**Healthy Weight**	20.0	23.7	0.7	22.3	23.2	23.5	24.1	24.9
		Overweight	25.0	26.8	1.1	25.0	26.0	26.9	27.7	28.5
		Obesity	14.0	33.0	2.0	30.2	31.8	32.8	33.8	37.8
Male	**19-39**	**Healthy Weight**	39.0	21.3	1.7	18.6	20.0	21.1	22.7	24.8
		Overweight	16.0	26.3	1.2	25.2	25.4	26.0	26.9	28.7
		Obesity	15.0	31.7	2.6	29.0	30.0	30.5	32.8	38.3
	40-59	**Healthy Weight**	54.0	22.3	1.6	18.5	20.8	22.2	23.6	24.9
		Overweight	30.0	27.1	1.1	25.1	26.2	27.0	28.1	28.8
		Obesity	33.0	32.0	1.9	29.2	30.6	31.6	32.7	36.6
	60+	**Healthy Weight**	13.0	22.5	1.6	20.0	21.0	22.5	24.0	24.6
		Overweight	14.0	26.6	1.1	25.6	25.8	26.2	27.4	28.9
		Obesity	13.0	31.3	2.3	29.3	29.8	30.1	32.0	37.0

In [57]:
```
# Underweight data
df_no_out[df_no_out['BMI']< 18.5].loc[:,['AGE','GENDER','BMI','BP','AGE_strata']]
```

Out[57]:

	AGE	GENDER	BMI	BP	AGE_strata
281	23	Female	18.0	78.0	19-39
381	29	Female	18.1	73.0	19-39

In the age group of 19-39 years old and females, we have only two individuals with a BMI categorized as underweight.

Question 3

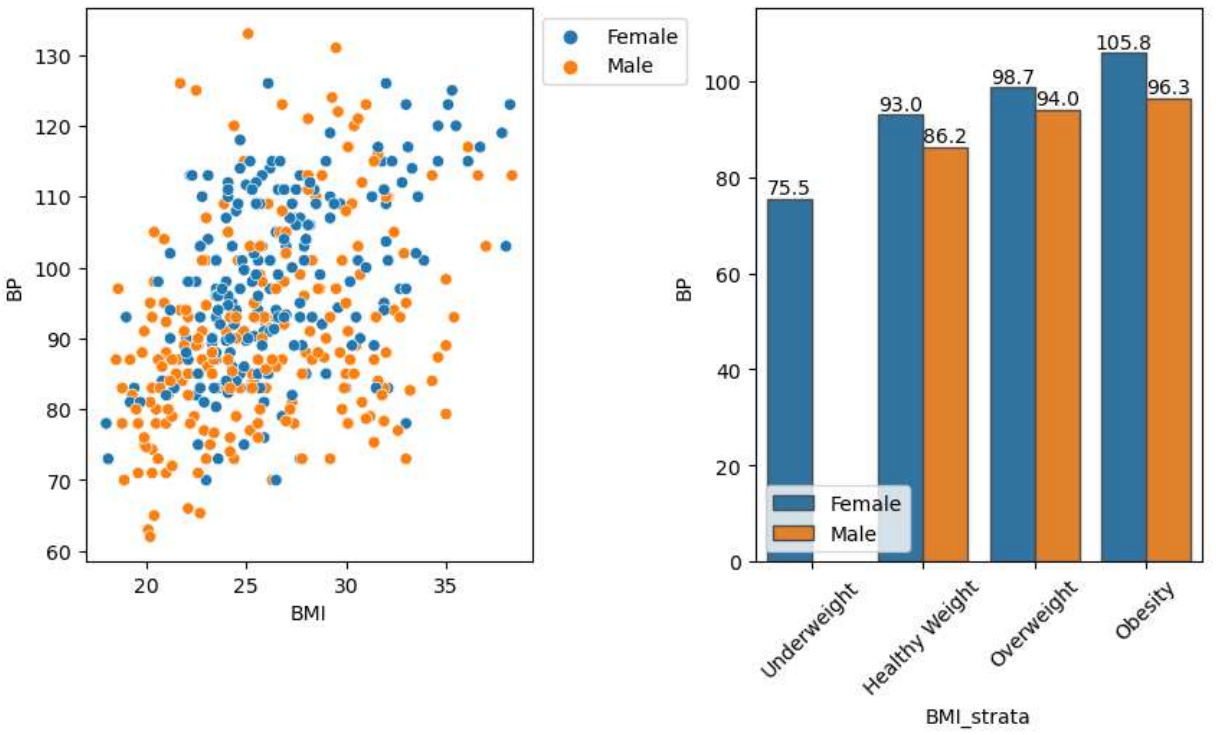
What is the relationship between **BMI** and **blood pressure**, and are there specific BMI categories associated with elevated or reduced blood pressure?

In [58]:
```
# Create 2 plots side by side of BMI vs AGE, and BMI vs Age groups:
fig, (ax5,ax6) = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

# 1st Plot
g5 = sns.scatterplot(data=df_no_out, x='BMI', y='BP', hue='GENDER', ax=ax5)
ax5.set(ylabel='BP')
g5.legend(loc='upper left',bbox_to_anchor=(1,1))

# 2nd Plot
g6 = sns.barplot(x='BMI_strata', y='BP', data=df_no_out, errwidth=0, hue='GENDER',
# add the annotation
for i in ax6.containers:
    ax6.bar_label(i,fmt='%.1f', label_type='edge')
ax6.set(ylabel='BP')
g6.legend(loc=3)
g6.set_xticklabels(labels=['Underweight', 'Healthy Weight','Overweight','Obesity'],

plt.subplots_adjust(wspace=0.5)   # Adjust the space between both plots
plt.show();
```



Standard BMI values for males and females per age groups:

BMI	Weight Status
below 18.5	Underweight
18.5 - 24.9	Healthy Weight
25.0 - 29.9	Overweight
30.0 and above	Obesity

- In every BMI weight group, females consistently exhibit higher blood pressure than males in our dataset.
- Another noteworthy observation from the data is the increase in blood pressure with an increase in BMI.
- Both males and females demonstrate that as weight (BMI) increases, blood pressure also increases, suggesting a positive relationship between blood pressure and weight. The previously calculated correlation coefficient for BMI and blood pressure indicated a medium correlation between them.

In [59]:
```
df_no_out.head()
```

Out[59]:

	AGE	GENDER	BMI	BP	AGE_strata	GENDERx	BMI_strata
0	59	Female	32.1	101.0	40-59	b	Obesity
1	48	Male	21.6	87.0	40-59	orange	Healthy Weight
2	72	Female	30.5	93.0	60+	b	Obesity
3	24	Male	25.3	84.0	19-39	orange	Overweight
4	50	Male	23.0	101.0	40-59	orange	Healthy Weight

In [60]:
```
# Summary statistics of BMI and blood pressure:
df_no_out.groupby(['BMI_strata']).describe()[['BMI','BP']].round(1)
```

Out[60]:

	BMI								BP				
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	2
BMI_strata													
Underweight	2.0	18.0	0.1	18.0	18.0	18.0	18.1	18.1	2.0	75.5	3.5	73.0	7
Healthy Weight	182.0	22.4	1.7	18.5	21.0	22.8	24.0	24.9	182.0	89.0	12.0	62.0	8
Overweight	134.0	26.7	1.1	25.0	25.7	26.6	27.7	28.9	134.0	96.6	12.2	70.0	8
Obesity	110.0	32.0	2.3	29.0	30.2	31.7	33.0	38.3	110.0	100.6	14.9	73.0	8

We observe how the BMI (weight) increases the blood pressure also does for each BMI classification, which clearly relate the rise of blood pressure with BMI (weight).

In [61]:
```
# Make a linear regreesion of BMI vs BP with gender and age as part of the multiple
# Another one, with age groups, and BMI classification
```

We will use multiple linear regression to find the line that best fits our data, it used to help us understand the relationships between BMI vs. blood pressure and age.

In [62]:
```
import statsmodels.api as sm
y = df_no_out[['BMI']]
x = df_no_out[['BP','AGE']]
x = sm.add_constant(x)
model = sm.OLS(y,x).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    BMI   R-squared:                       0.179
Model:                            OLS   Adj. R-squared:                  0.175
Method:                 Least Squares   F-statistic:                     46.25
Date:                Thu, 28 Dec 2023   Prob (F-statistic):           6.73e-19
Time:                        13:32:18   Log-Likelihood:                -1184.5
No. Observations:                 428   AIC:                             2375.
Df Residuals:                     425   BIC:                             2387.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         13.6110      1.326     10.267      0.000      11.005      16.217
BP             0.1201      0.014      8.362      0.000       0.092       0.148
AGE            0.0268      0.015      1.780      0.076      -0.003       0.056
==============================================================================
Omnibus:                       21.505   Durbin-Watson:                   2.127
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               23.640
Skew:                           0.564   Prob(JB):                     7.36e-06
Kurtosis:                       2.765   Cond. No.                         760.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

The coefficient of determination (r-squared): it is the proportion of the variance in the response variable that can be explained by the predictor variable. In our case, r-squared is 0.179 indicating that 18% of the variance in the response variable (BMI) can be explained by the predictor variable (BP and Age). 18% of variability in BMI can be explained by the predictor variable (BP and Age). This tells us how good is BP and Age predictor of BMI. 18% of the variation in BMI can be explained by BP and Age.

The p_value for Age is 0.076, which is bigger than our significance level alpha of 0.05. This is not statistically significant. Since Age is not statistically significant, we may end up deciding to remove it from the model.

Prob(F-statistic)=6.73e-19, it is the p_value associated with the overall F-statistic. It tells us whether or not the regression model as a whole is statistically significant. It tells us if the two preditor variables (BP and Age) combined have a statistically significant association with the response variable (BMI).

In this case, the p_value is less than 0.05, which indicates that the predictor variables BP and Age combined have a statistically significant association with BMI.

This model performed the Durbin-watson test to check the independence of the residuals, its value is 2.127 This test assumes that there is no correlation between the residuals, in other words, the residuals are assumed to be independent.The result indicates no correlation among the residuals, its value is considered normal.

This model performed the Jarque-Bera test to check the normality of residuals, its value is 23.640. It is a goodness-of-fit test that determines whether or not sample data have skewness and kurtosis that matches a normal distribution. The further it is from zero, the more evidence that the sample data doesn't follow a normal distribution.

In [63]:
```
# Jaque-Bera Test
import scipy.stats as stats
data=df_no_out[['BMI','BP','AGE']]
stats.jarque_bera(data)
```

Out[63]:
```
SignificanceResult(statistic=116.8935571633388, pvalue=4.13891566214991e-26)
```

Hypothesis

"There is a significant gender-based difference in the average BMI among adults." You can test this hypothesis using a two-sample t-test to compare the mean BMI of males and females. This study can help determine whether gender is associated with differences in BMI and whether specific gender-based health interventions are necessaryonship.

In [64]:
```
df_no_out.groupby('GENDER')[['BMI','BP']].mean().round(2)
```

Out[64]:

GENDER	BMI	BP
Female	26.64	98.02
Male	25.86	90.96

Let's make a Welch's t-test for the mean of BMI of females and males in our dataset
x1 = sample females BMI mean
x2 = sample males BMI mean

Null Hypothesis: H0: x1 = x2
Alternative Hypothesis: Ha: x1 > x2

In [65]:
```
# Female random sample
df_bmi_f = df_no_out.loc[df_no_out['GENDER']=='Female', 'BMI']
x1_sample = df_bmi_f.take(np.random.permutation(len(df_bmi_f))[:5])
x1_sample
```

Out[65]:
```
49     27.7
229    24.9
368    26.6
109    25.5
17     27.5
Name: BMI, dtype: float64
```

In [66]:
```
# Male random sample
df_bmi_m = df_no_out.loc[df_no_out['GENDER']=='Male', 'BMI']
x2_sample = df_bmi_m.take(np.random.permutation(len(df_bmi_m))[:5])
x2_sample
```

Out[66]:
```
107    31.0
112    28.3
190    25.2
412    34.3
252    31.9
Name: BMI, dtype: float64
```

In [67]:
```
df_no_out.groupby('GENDER').mean()['BMI'].round(2)
```

Out[67]:
```
GENDER
Female    26.64
Male      25.86
Name: BMI, dtype: float64
```

In [68]:
```
x1_mean = round(df_no_out.loc[df_no_out['GENDER']=='Female', 'BMI'].mean(),2)
x2_mean = round(df_no_out.loc[df_no_out['GENDER']=='Male', 'BMI'].mean(),2)
print(f'Female average = {x1_mean}')
print(f'Male average = {x2_mean}')
```

```
Female average = 26.64
Male average = 25.86
```

In [69]:
```
# Female and Male variance of BMI
var1_bmi = round(df_bmi_f.var(),2)
var2_bmi = round(df_bmi_m.var(),2)
print(var1_bmi)
print(var2_bmi)
```

```
15.96
19.8
```

In [70]:
```
# Sample size for Females and Males BMI
n1_bmi = df_bmi_f.size
n2_bmi = df_bmi_m.size
print(n1_bmi)
print(n2_bmi)
```

```
201
227
```

In [71]:
```
# Welch's t-test
x1_bmi = 26.64
x2_bmi = 25.86
var1_bmi
var2_bmi
```

```
n1_bmi
n2_bmi

welch_bmi = round((x1_bmi - x2_bmi) / np.sqrt((var1_bmi/n1_bmi) + (var2_bmi/n2_bmi)
welch_bmi
```

Out[71]: 1.9108

In [72]:
```
degree_of_freedom_bmi = int(round(((var1_bmi/n1_bmi) + (var2_bmi/n2_bmi))**2 / ((va
degree_of_freedom_bmi
```

Out[72]: 426

According to the tables, the t-critical is 1.966 for a significant level of 5%, degree of freedom of 426, and two-sample test.

The analysis involving the t-critical value and Welch's test statistic supports the rejection of the null hypothesis. This suggests that there is a significant difference in the mean BMI between females and males. The alternative hypothesis is deemed true, indicating that the average BMI for females is higher than that for males.

Conclusions

- Data analysis by gender reveals higher BMI and Blood Pressure values in females compared to males.
- The analysis highlights the impact of age on Blood Pressure and BMI, indicating an increase with advancing age. On average, we have lower Blood Pressure in males.
- The age group of 40 to 59 years exhibits the highest maximum Blood Pressure values with a broader range than other age groups.
- Differences in blood pressure among age groups and genders are significantly pronounced in the youngest ages.
- Males consistently have lower BMI than females in each age group, aligning with the trend observed for blood pressure vs. age.
- Females, on average, tend to have a higher BMI than males, displaying a consistent pattern.

- Comparing BMI in the dataset with CDC standards reveals that only males in the 19-39 age group fall within the Healthy Weight segment, while others are considered overweight.
- In the 19-39 age group for females, only two individuals have a BMI categorized as underweight, indicating rarity in this category.
- There is a noticeable positive relationship between blood pressure and BMI, with an increase in BMI associated with an increase in blood pressure for both genders.
- Tests indicate that blood pressure and age are moderately good predictors of BMI, explaining a portion of the variability, and have a statistically significant association with

BMI.