medical-premium-insurance-project

November 16, 2023

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

[2]:
```
#IMPORTING THE DATA
```

[3]:
```
dt = pd.read_csv("C:/Users/Adithya Das/Desktop/Medicalpremium.csv")
```

[4]:
```
#DATA ANALYSIS
```

[5]:
```
dt.shape
```

[5]:
```
(986, 11)
```

[6]:
```
dt.head()
```

[6]:
```
   Age  Diabetes  BloodPressureProblems  AnyTransplants  AnyChronicDiseases  \
0   45         0                      0               0                   0
1   60         1                      0               0                   0
2   36         1                      1               0                   0
3   52         1                      1               0                   1
4   38         0                      0               0                   1

   Height  Weight  KnownAllergies  HistoryOfCancerInFamily  \
0     155      57               0                        0
1     180      73               0                        0
2     158      59               0                        0
3     183      93               0                        0
4     166      88               0                        0

   NumberOfMajorSurgeries  PremiumPrice
0                       0         25000
1                       0         29000
```

```
2                        1         23000
3                        2         28000
4                        1         23000
```

[7]:
```
dt.tail()
```

[7]:
```
     Age  Diabetes  BloodPressureProblems  AnyTransplants  AnyChronicDiseases  \
981   18         0                      0               0                   0
982   64         1                      1               0                   0
983   56         0                      1               0                   0
984   47         1                      1               0                   0
985   21         0                      0               0                   0

     Height  Weight  KnownAllergies  HistoryOfCancerInFamily  \
981     169      67               0                        0
982     153      70               0                        0
983     155      71               0                        0
984     158      73               1                        0
985     158      75               1                        0

     NumberOfMajorSurgeries  PremiumPrice
981                       0         15000
982                       3         28000
983                       1         29000
984                       1         39000
985                       1         15000
```

[8]:
```
dt.describe()
```

[8]:
```
              Age    Diabetes  BloodPressureProblems  AnyTransplants  \
count  986.000000  986.000000             986.000000      986.000000
mean    41.745436    0.419878               0.468560        0.055781
std     13.963371    0.493789               0.499264        0.229615
min     18.000000    0.000000               0.000000        0.000000
25%     30.000000    0.000000               0.000000        0.000000
50%     42.000000    0.000000               0.000000        0.000000
75%     53.000000    1.000000               1.000000        0.000000
max     66.000000    1.000000               1.000000        1.000000

       AnyChronicDiseases      Height      Weight  KnownAllergies  \
count          986.000000  986.000000  986.000000      986.000000
mean             0.180527  168.182556   76.950304        0.215010
std              0.384821   10.098155   14.265096        0.411038
min              0.000000  145.000000   51.000000        0.000000
25%              0.000000  161.000000   67.000000        0.000000
50%              0.000000  168.000000   75.000000        0.000000
75%              0.000000  176.000000   87.000000        0.000000
```

```
max              1.000000  188.000000  132.000000        1.000000

       HistoryOfCancerInFamily  NumberOfMajorSurgeries  PremiumPrice
count               986.000000              986.000000    986.000000
mean                  0.117647                0.667343  24336.713996
std                   0.322353                0.749205   6248.184382
min                   0.000000                0.000000  15000.000000
25%                   0.000000                0.000000  21000.000000
50%                   0.000000                1.000000  23000.000000
75%                   0.000000                1.000000  28000.000000
max                   1.000000                3.000000  40000.000000
```

[9]:

```
#CHECK FOR NULL VALUE
```

[10]:

```
dt.isnull().sum()
```

[10]:

```
Age                        0
Diabetes                   0
BloodPressureProblems      0
AnyTransplants             0
AnyChronicDiseases         0
Height                     0
Weight                     0
KnownAllergies             0
HistoryOfCancerInFamily    0
NumberOfMajorSurgeries     0
PremiumPrice               0
dtype: int64
```

[11]:

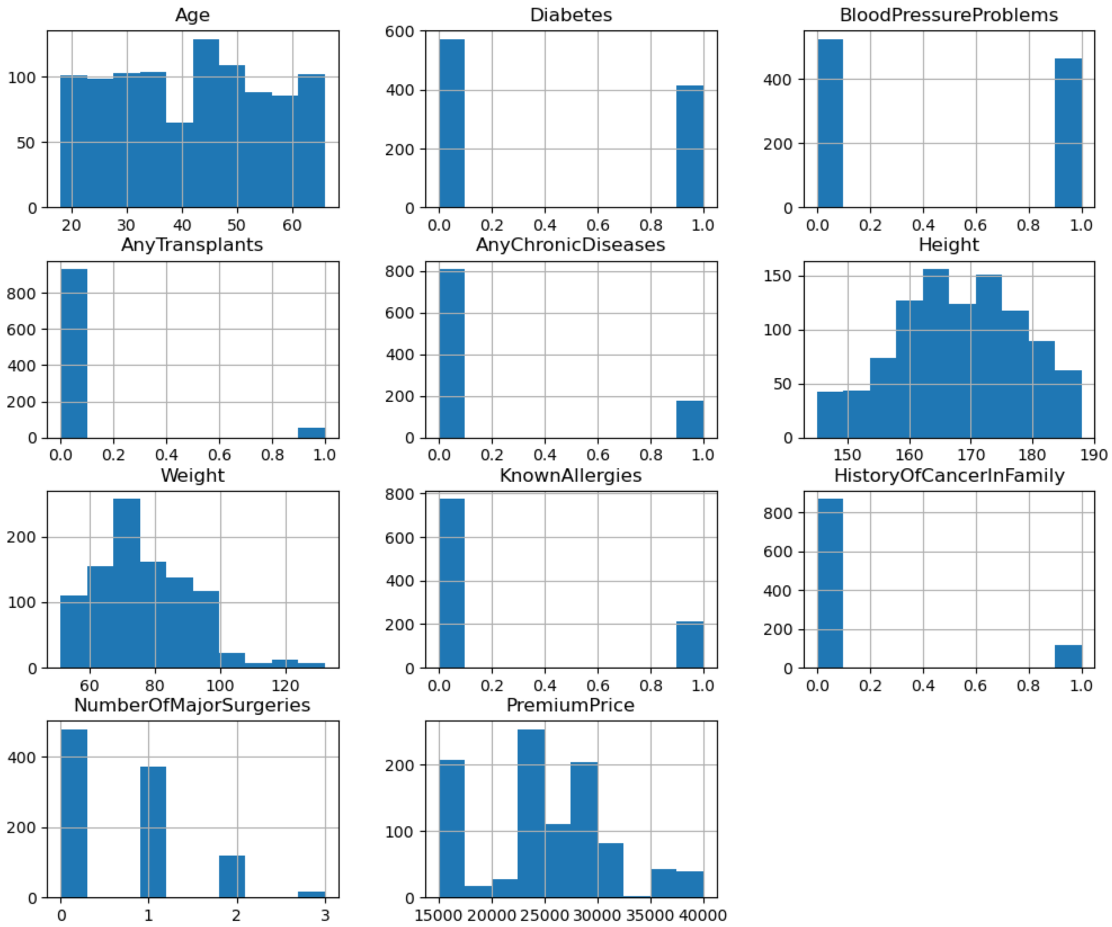
```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 986 entries, 0 to 985
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Age                      986 non-null    int64
 1   Diabetes                 986 non-null    int64
 2   BloodPressureProblems    986 non-null    int64
 3   AnyTransplants           986 non-null    int64
 4   AnyChronicDiseases       986 non-null    int64
 5   Height                   986 non-null    int64
 6   Weight                   986 non-null    int64
 7   KnownAllergies           986 non-null    int64
 8   HistoryOfCancerInFamily  986 non-null    int64
 9   NumberOfMajorSurgeries   986 non-null    int64
 10  PremiumPrice             986 non-null    int64
```

```
dtypes: int64(11)
memory usage: 84.9 KB
```

[12]:

```
dt.hist(figsize=(12, 10))
plt.show()
```



[13]:

```
#CATEGORICAL DATA ANALYSIS
```

[14]:

```
dt.groupby('Diabetes')['PremiumPrice'].mean()
dt.groupby('BloodPressureProblems')['PremiumPrice'].mean()
```

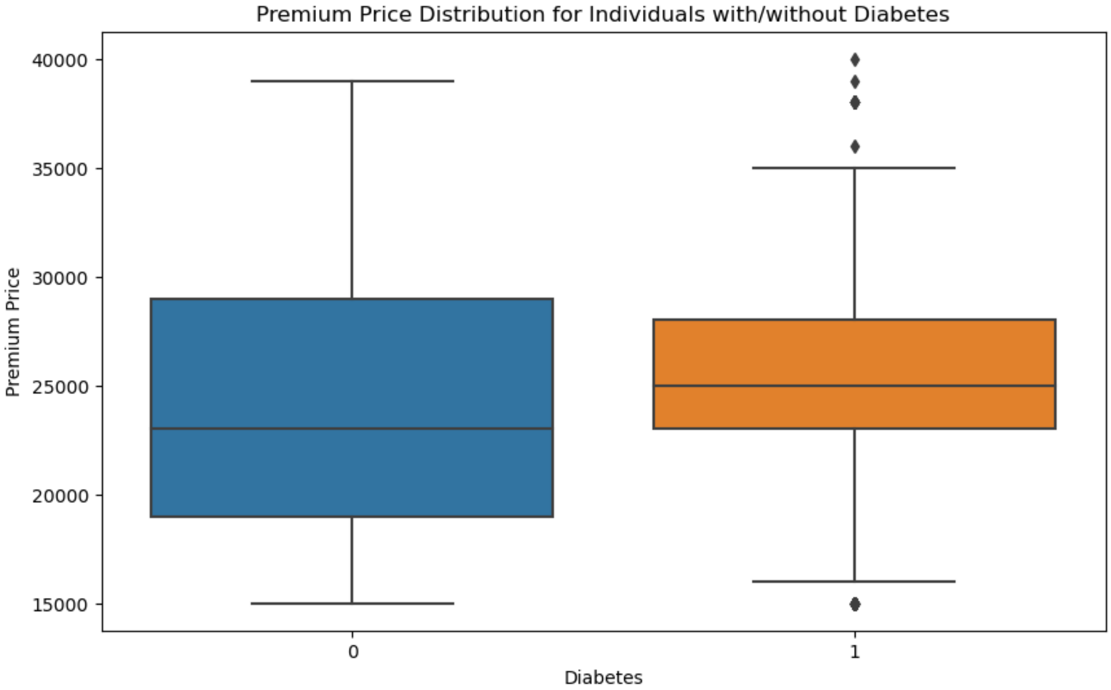
[14]:

```
BloodPressureProblems
0    23356.870229
1    25448.051948
Name: PremiumPrice, dtype: float64
```

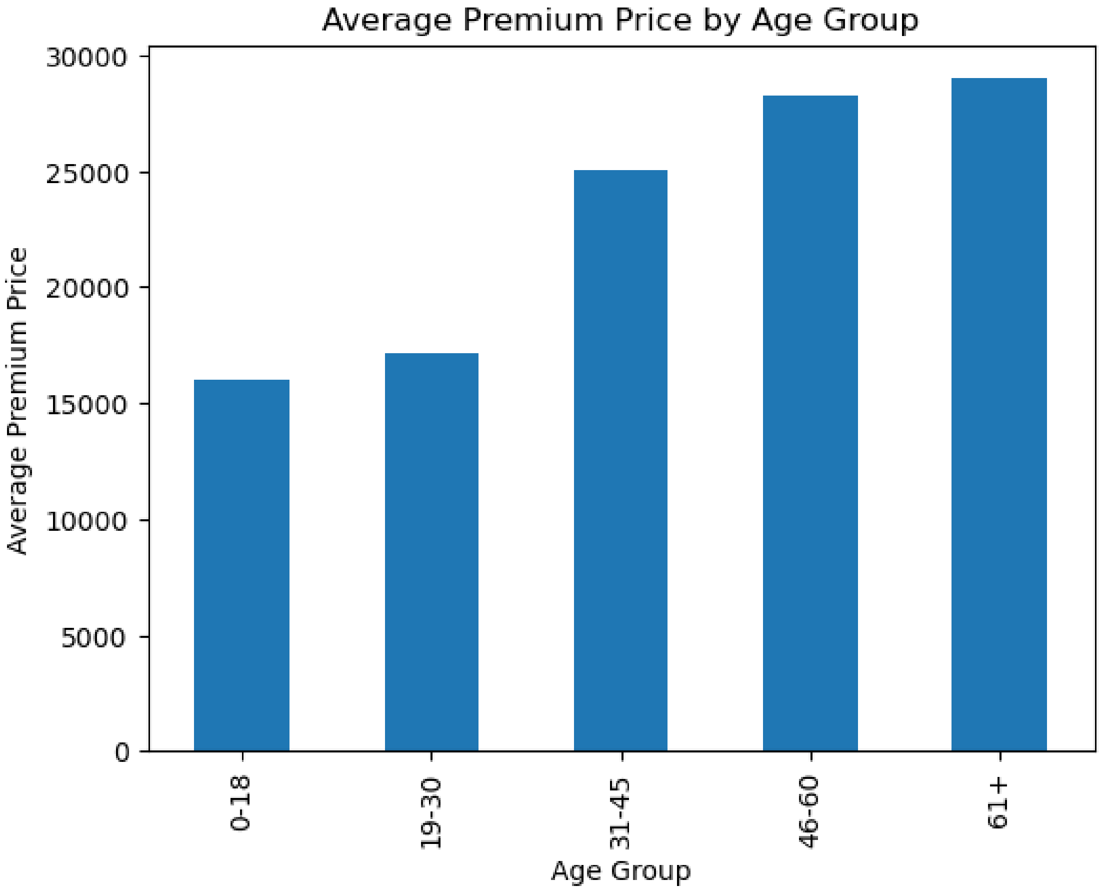
[15]:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Diabetes', y='PremiumPrice', data=dt)
```

```
plt.xlabel('Diabetes')
plt.ylabel('Premium Price')
plt.title('Premium Price Distribution for Individuals with/without Diabetes')
plt.show()
```

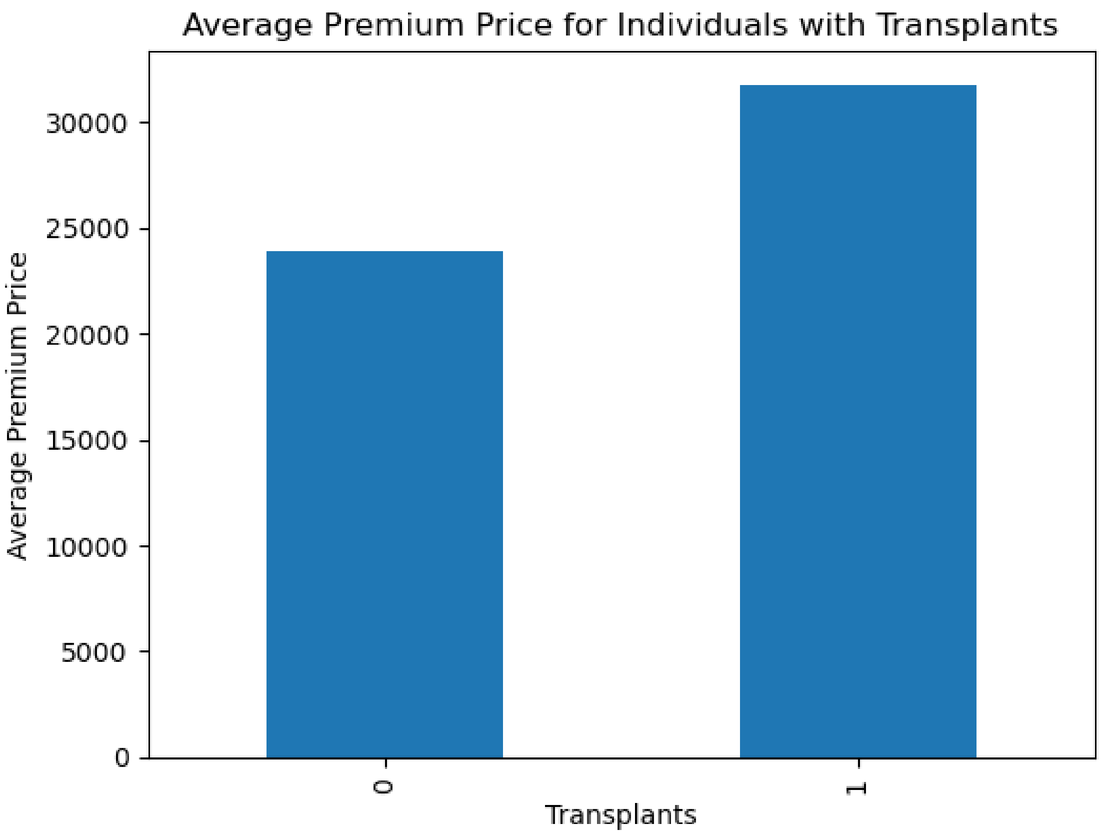


[16]:
```
age_bins = [0, 18, 30, 45, 60, 100]
age_labels = ['0-18', '19-30', '31-45', '46-60', '61+']
dt['AgeGroup'] = pd.cut(dt['Age'], bins=age_bins, labels=age_labels)
dt.groupby('AgeGroup')['PremiumPrice'].mean().plot(kind='bar')
plt.xlabel('Age Group')
plt.ylabel('Average Premium Price')
plt.title('Average Premium Price by Age Group')
plt.show()
```



[17]:

```
dt.groupby('AnyTransplants')['PremiumPrice'].mean().plot(kind='bar')
plt.xlabel('Transplants')
plt.ylabel('Average Premium Price')
plt.title('Average Premium Price for Individuals with Transplants')
plt.show()
```

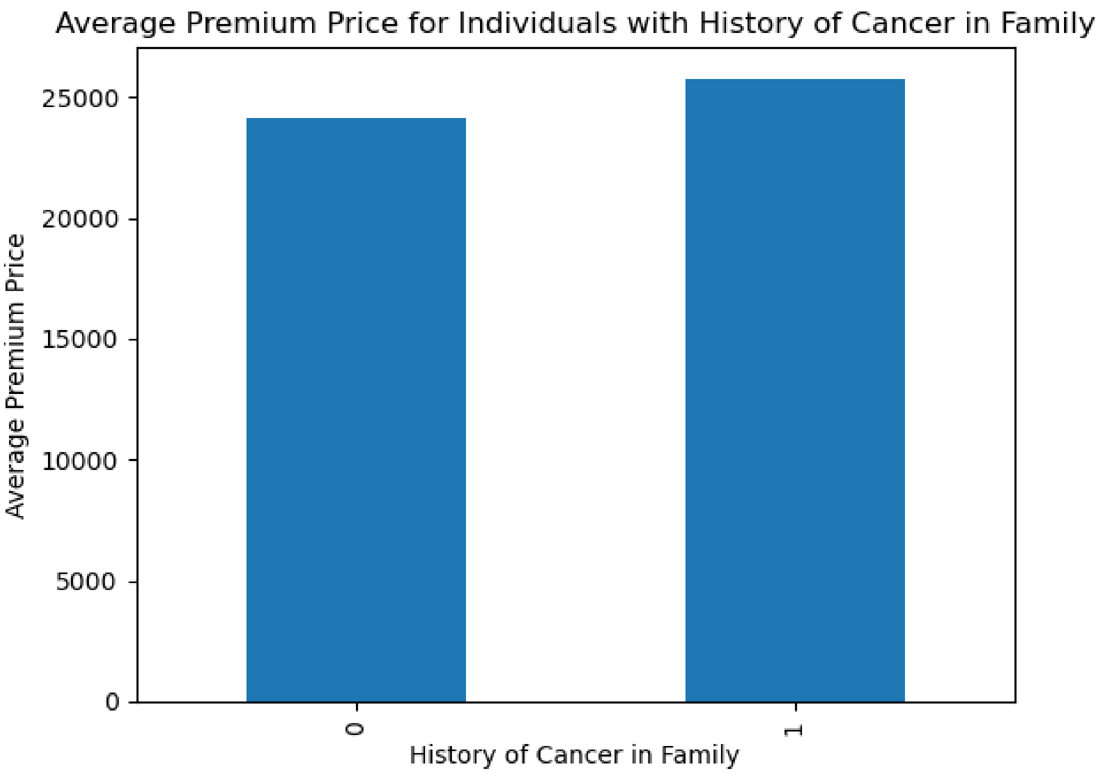


[18]:

```
#HISTORY ANALYSIS
```

[19]:

```
dt.groupby('HistoryOfCancerInFamily')['PremiumPrice'].mean().plot(kind='bar')
plt.xlabel('History of Cancer in Family')
plt.ylabel('Average Premium Price')
plt.title('Average Premium Price for Individuals with History of Cancer in␣
  ↪Family')
plt.show()
```

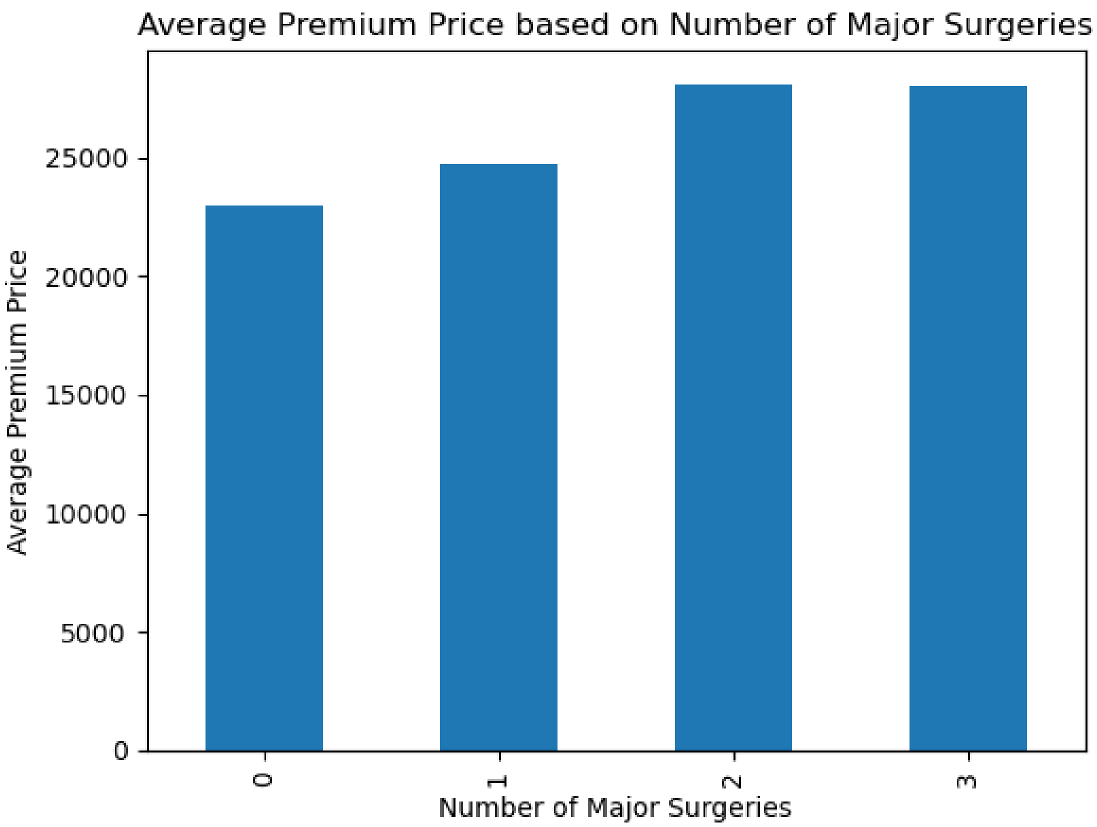


[20]:

```
#NO. OF SURGERY ANALYSIS
```

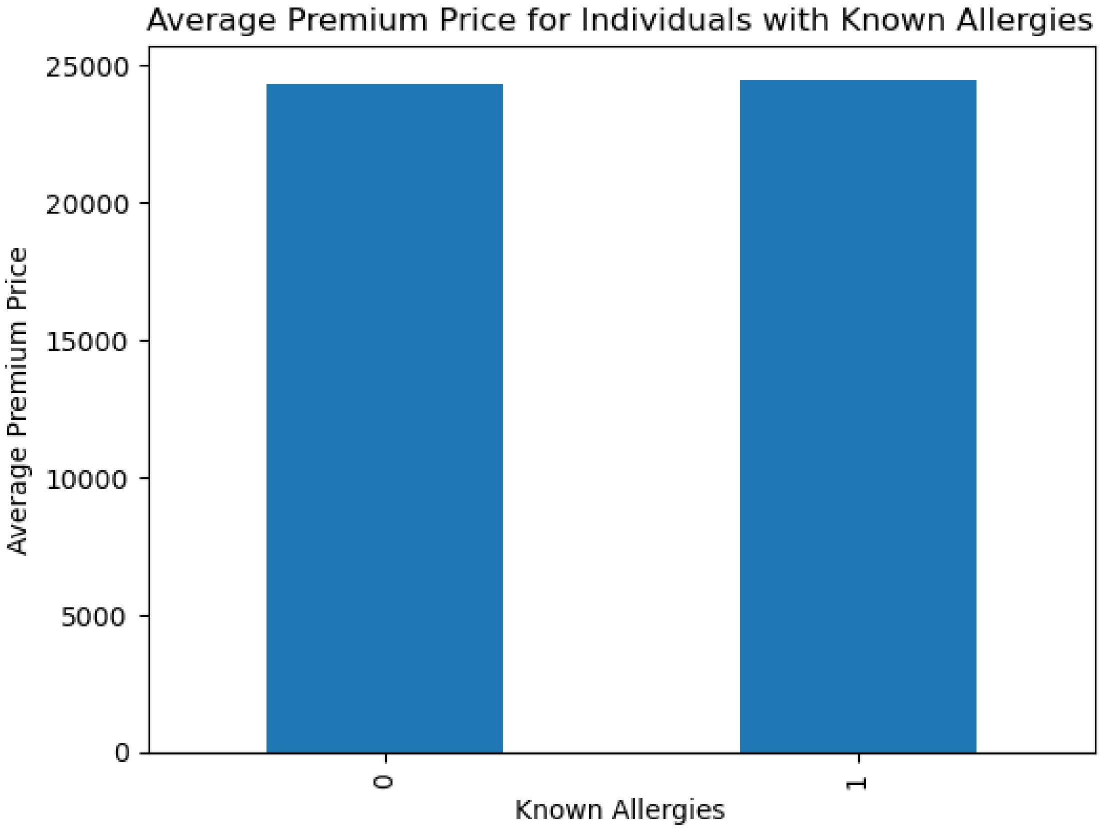
[21]:

```
dt.groupby('NumberOfMajorSurgeries')['PremiumPrice'].mean().plot(kind='bar')
plt.xlabel('Number of Major Surgeries')
plt.ylabel('Average Premium Price')
plt.title('Average Premium Price based on Number of Major Surgeries')
plt.show()
```



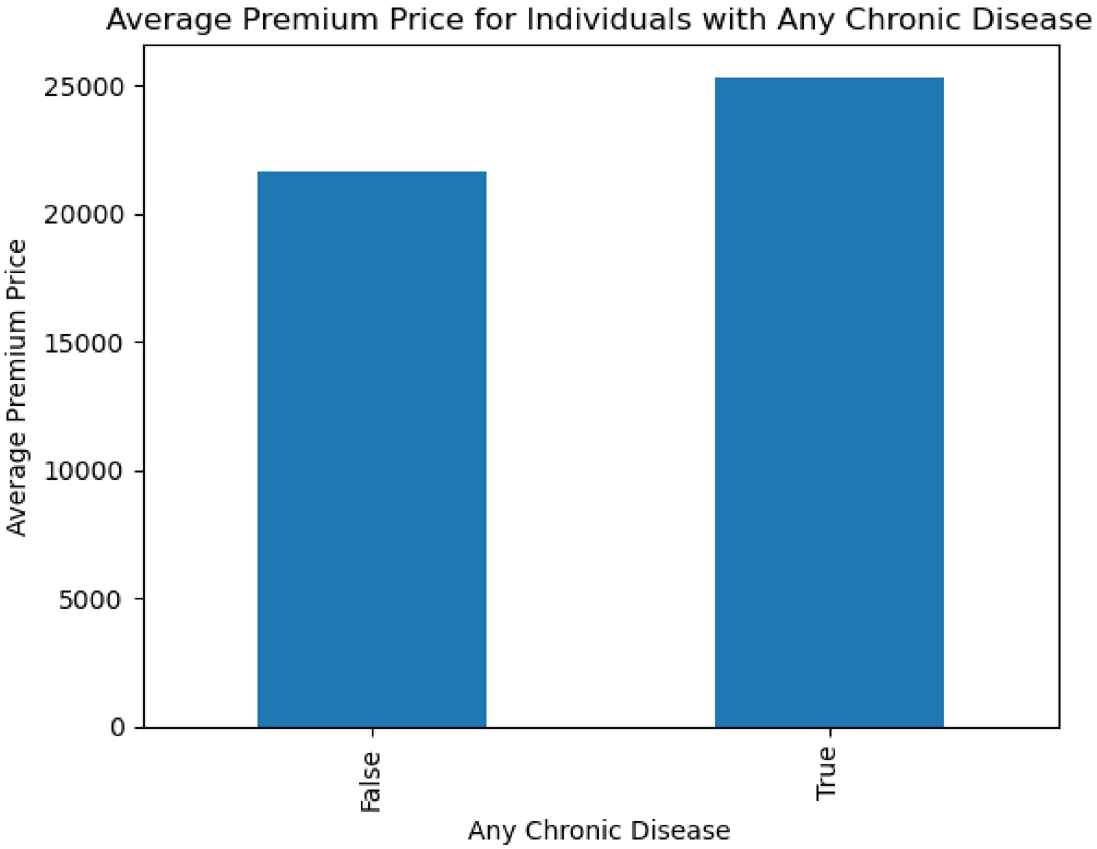
```
[22]: #KNOWN ALLERGIES ANALYSIS
```

```
[23]: dt.groupby('KnownAllergies')['PremiumPrice'].mean().plot(kind='bar')
      plt.xlabel('Known Allergies')
      plt.ylabel('Average Premium Price')
      plt.title('Average Premium Price for Individuals with Known Allergies')
      plt.show()
```



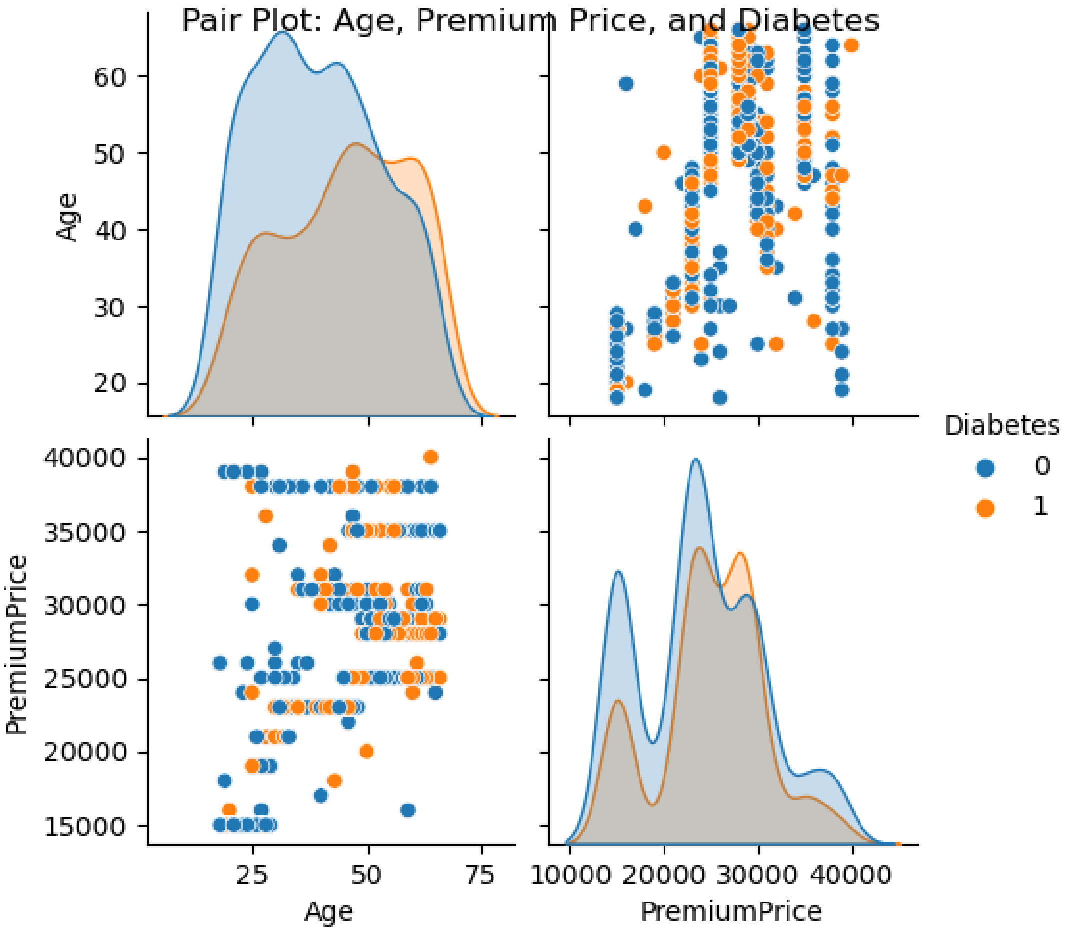
[24]:

```python
chronic_diseases = ['Diabetes', 'BloodPressureProblems', 'AnyChronicDiseases']
dt['AnyChronicDisease'] = dt[chronic_diseases].any(axis=1)
dt.groupby('AnyChronicDisease')['PremiumPrice'].mean().plot(kind='bar')
plt.xlabel('Any Chronic Disease')
plt.ylabel('Average Premium Price')
plt.title('Average Premium Price for Individuals with Any Chronic Disease')
plt.show()
```



[25]:

```
sns.pairplot(dt, vars=['Age', 'PremiumPrice'], hue='Diabetes', diag_kind='kde')
plt.suptitle('Pair Plot: Age, Premium Price, and Diabetes')
plt.show()
```



```
[26]: #CORRELATION
```

```
[27]: dt.corr()
```

```
C:\Users\Adithya Das\AppData\Local\Temp\ipykernel_8652\1921767823.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  dt.corr()
```

```
[27]:                              Age  Diabetes  BloodPressureProblems  \
      Age                     1.000000  0.210908               0.244888
      Diabetes                0.210908  1.000000               0.127727
      BloodPressureProblems   0.244888  0.127727               1.000000
      AnyTransplants         -0.008549 -0.036652              -0.024538
      AnyChronicDiseases      0.051072 -0.089428               0.045424
      Height                  0.039879 -0.003783              -0.037926
      Weight                 -0.018590 -0.024563              -0.061016
```

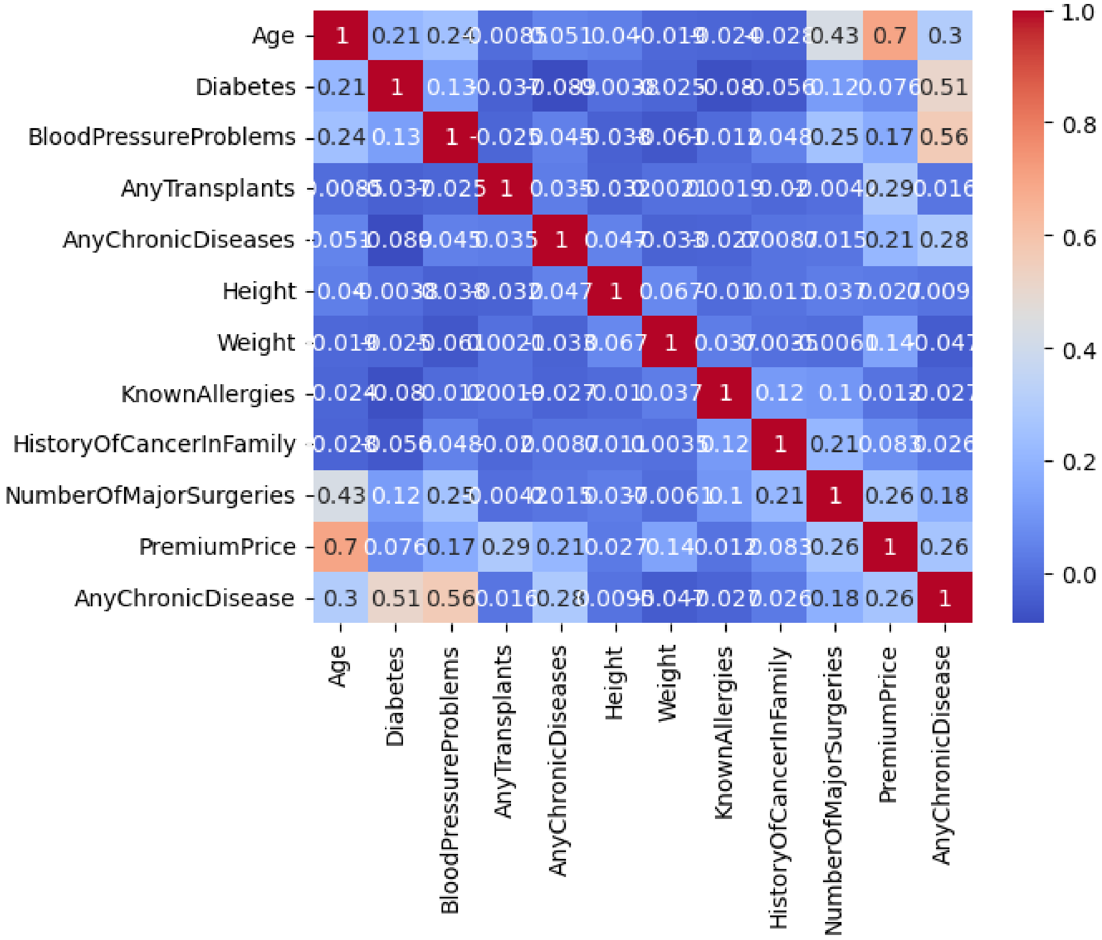
```
KnownAllergies          -0.024416 -0.080102              -0.011550
HistoryOfCancerInFamily -0.027623 -0.055527               0.048239
NumberOfMajorSurgeries    0.429181  0.122722               0.251568
PremiumPrice              0.697540  0.076209               0.167097
AnyChronicDisease         0.300208  0.510450               0.563387

                         AnyTransplants  AnyChronicDiseases    Height  \
Age                           -0.008549            0.051072  0.039879
Diabetes                      -0.036652           -0.089428 -0.003783
BloodPressureProblems         -0.024538            0.045424 -0.037926
AnyTransplants                 1.000000            0.035285 -0.031543
AnyChronicDiseases             0.035285            1.000000  0.047419
Height                        -0.031543            0.047419  1.000000
Weight                         0.002087           -0.033318  0.066946
KnownAllergies                 0.001876           -0.027418 -0.010200
HistoryOfCancerInFamily       -0.020171            0.008666  0.010549
NumberOfMajorSurgeries        -0.004154            0.014835  0.037289
PremiumPrice                   0.289056            0.208610  0.026910
AnyChronicDisease              0.015615            0.281615  0.009486

                            Weight  KnownAllergies  HistoryOfCancerInFamily  \
Age                      -0.018590       -0.024416                -0.027623
Diabetes                 -0.024563       -0.080102                -0.055527
BloodPressureProblems    -0.061016       -0.011550                 0.048239
AnyTransplants            0.002087        0.001876                -0.020171
AnyChronicDiseases       -0.033318       -0.027418                 0.008666
Height                    0.066946       -0.010200                 0.010549
Weight                    1.000000        0.037492                 0.003481
KnownAllergies            0.037492        1.000000                 0.115383
HistoryOfCancerInFamily   0.003481        0.115383                 1.000000
NumberOfMajorSurgeries   -0.006108        0.103923                 0.212657
PremiumPrice              0.141507        0.012103                 0.083139
AnyChronicDisease        -0.047237       -0.027320                 0.026442

                         NumberOfMajorSurgeries  PremiumPrice  \
Age                                    0.429181      0.697540
Diabetes                               0.122722      0.076209
BloodPressureProblems                  0.251568      0.167097
AnyTransplants                        -0.004154      0.289056
AnyChronicDiseases                     0.014835      0.208610
Height                                 0.037289      0.026910
Weight                                -0.006108      0.141507
KnownAllergies                         0.103923      0.012103
HistoryOfCancerInFamily                0.212657      0.083139
NumberOfMajorSurgeries                 1.000000      0.264250
PremiumPrice                           0.264250      1.000000
AnyChronicDisease                      0.184738      0.260210
```

```
                         AnyChronicDisease
Age                               0.300208
Diabetes                          0.510450
BloodPressureProblems             0.563387
AnyTransplants                    0.015615
AnyChronicDiseases                0.281615
Height                            0.009486
Weight                           -0.047237
KnownAllergies                   -0.027320
HistoryOfCancerInFamily           0.026442
NumberOfMajorSurgeries            0.184738
PremiumPrice                      0.260210
AnyChronicDisease                 1.000000
```

[28]:

```
sns.heatmap(dt.corr(), annot=True, cmap='coolwarm')
plt.show()
```

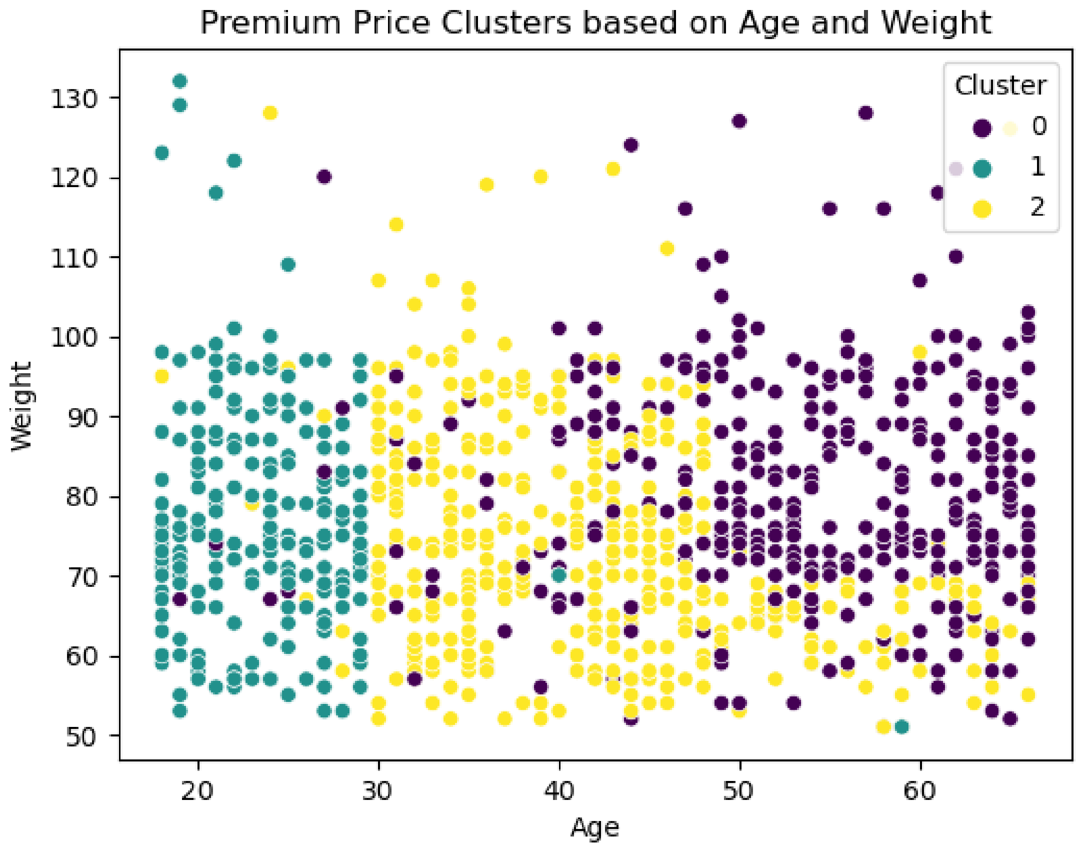
```
C:\Users\Adithya Das\AppData\Local\Temp\ipykernel_8652\1692465306.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(dt.corr(), annot=True, cmap='coolwarm')
```



[29]:
```
from sklearn.cluster import KMeans

# Prepare features
X = dt[['Age', 'Weight', 'PremiumPrice']]

# Use KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
dt['Cluster'] = kmeans.fit_predict(X)

# Visualize clusters
sns.scatterplot(data=dt, x='Age', y='Weight', hue='Cluster', palette='viridis')
plt.xlabel('Age')
plt.ylabel('Weight')
plt.title('Premium Price Clusters based on Age and Weight')
plt.show()
```

```
C:\Users\Adithya Das\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
```

```
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Adithya Das\anaconda3\Lib\site-
packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a
memory leak on Windows with MKL, when there are less chunks than available
threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=4.
  warnings.warn(
```



```
[30]: from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_squared_error

      # Prepare features and target variable
      features = ['Age', 'Diabetes', 'BloodPressureProblems', 'AnyChronicDiseases',␣
        ↪'Height', 'Weight', 'KnownAllergies', 'HistoryOfCancerInFamily',␣
        ↪'NumberOfMajorSurgeries']
      X = dt[features]
      y = dt['PremiumPrice']

      # Split data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Build and train the regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)

# Calculate and print mean squared error
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')
```

```
Mean Squared Error: 16775137.514773391
```

[ ]:

[ ]:

[ ]: