UNIVERSITY OF LIVERPOOL

INDIVIDUAL REPORT

Module: ACFI827 - Introduction to Programming (Python)

Submitted to

Dr Swati Sachan

Lecturer in Artificial Intelligence in Finance

University of Liverpool Management School

Submitted by

Daphne Nguyen

ID: 201775118

MSc in Financial Technology

University of Liverpool Management School

JANUARY 2024

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

I. TASK 1

Once the .csv dataset is imported in Python with Pandas, the first 8 rows of DataFrame (df) are shown by using the syntax df.head(8). The result is as follows:

	age	gender	bmi	...	occupation	coverage_level	charges
0	46	male	21.45	...	Blue collar	Premium	20460.307669
1	25	female	25.38	...	White collar	Premium	20390.899218
2	38	male	44.88	...	Blue collar	Premium	20204.476302
3	25	male	19.89	...	White collar	Standard	11789.029843
4	49	male	38.21	...	White collar	Standard	19268.309838
5	55	female	36.41	...	Student	Basic	11896.836613
6	64	female	20.12	...	Blue collar	Basic	9563.655011
7	53	male	30.51	...	Student	Standard	15845.293730

[8 rows x 12 columns]

Table 1: First 8 rows of the insurance df (PyCharm, 2023)

df.shape will count the total number of rows and columns of the df to find out the shape of the datasets.

```
(1000000, 12)
The insurance data-set contains 1000000 rows and 12 columns
```

Figure 1: Shape of the dataset (PyCharm, 2023)

II. TASK 2

2.1. Firstly, the number of missing values in each column is identified with df.isna().sum(). The output indicates that there are no missing values in all columns. Therefore, no technique is needed to handle insufficient data at this point.

```
age                       0
gender                    0
bmi                       0
children                  0
smoker                    0
region                    0
medical_history           0
family_medical_history    0
exercise_frequency        0
occupation                0
coverage_level            0
charges                   0
dtype: int64
```

Figure 2: Result of missing values in each variable (PyCharm, 2023)

2.2. The fundamental descriptive statistics of numerical columns, including Age, BMI, Children, and Charges are calculated by the input df.describe().

	age	bmi	children	charges
count	1000000.000000	1000000.000000	1000000.000000	1000000.000000
mean	41.495282	34.001839	2.499886	16735.117481
std	13.855189	9.231680	1.707679	4415.808211
min	18.000000	18.000000	0.000000	3445.011643
25%	29.000000	26.020000	1.000000	13600.372379
50%	41.000000	34.000000	2.000000	16622.127973
75%	53.000000	41.990000	4.000000	19781.465410
max	65.000000	50.000000	5.000000	32561.560374

Table 2: Statistical features of the numerical variables (PyCharm, 2023)

2.3. Similarly, to show the results of non-numerical columns, the data types of “Object” which contains general text or mixed data and “Boolean” values are added to the include parameter. The syntax used is df.describe(include=["object","bool"]).

	gender	smoker	region	medical_history	family_medical_history	\
count	1000000	1000000	1000000	1000000	1000000	
unique	2	2	4	4	4	
top	male	yes	northeast	None	None	
freq	500107	500129	250343	250762	250404	

	exercise_frequency	occupation	coverage_level
count	1000000	1000000	1000000
unique	4	4	3
top	Rarely	Unemployed	Basic
freq	250538	250571	333515

Table 3: Statistical details of the categorical columns (Jupyter Notebook, 2023)

III. TASK 3

3.1. The distribution of policyholders across 4 regions is illustrated in the following graphs:

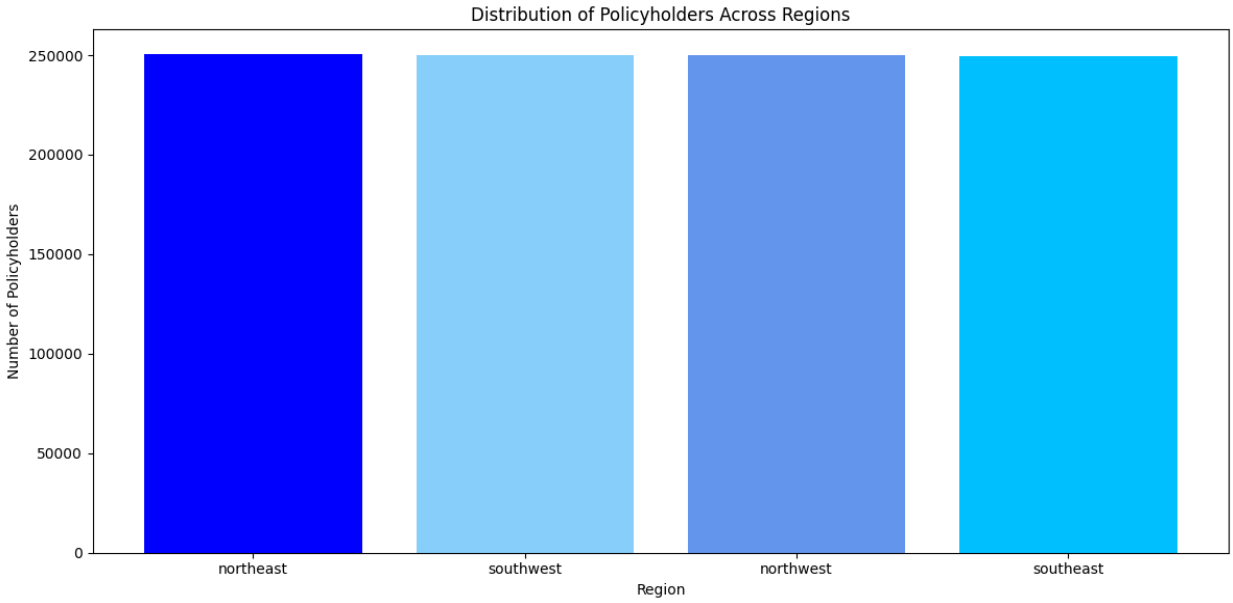

Figure 3: Number of policyholders in 4 regions in a Bar chart (Matplotlib, 2023)

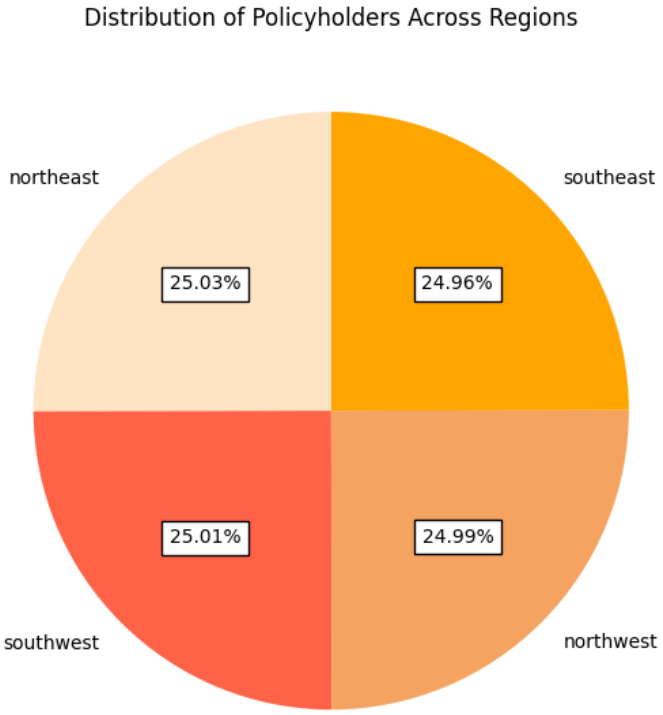

Figure 4: Percentages of policyholders in 4 regions in a Pie chart (Matplotlib, 2023)

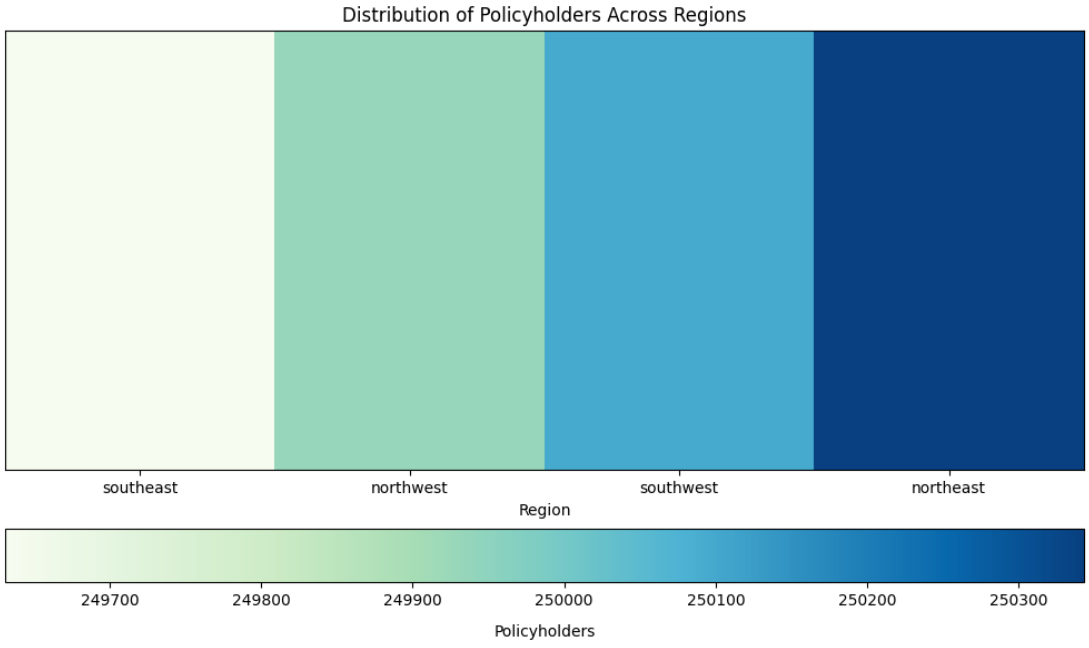

Figure 5: Heat map of the number of policyholders in 4 regions (Matplotlib, 2023)

It can be seen that the number of policyholders is distributed fairly equally across 4 regions. Among the three graphs, the heat map and pie chart are more descriptive than the bar chart as they indicate the specific numbers of people in 4 areas. While the pie chart only shows the proportions of people from all regions, the heat map ranks them in increasing order. In conclusion, most policyholders in this insurance dataset reside in the Northeast region with 250,300 people, whereas the Southeast area is the least with 249,700 people.

3.2. Before calculating the correlation of all features, the categorical columns will be assigned dummy variables using the pd.get_dummies function (Tyshevskyi, P 2019). Then the correlation matrix between all columns is computed with the Pearson method, which is df.corr(). The result is as below:

	age	...	coverage_level_Standard
age	1.000000	...	-0.000055
bmi	0.001428	...	-0.001014
children	-0.001317	...	0.001015
charges	0.063390	...	-0.051592
gender_female	0.001066	...	-0.000599
gender_male	-0.001066	...	0.000599
smoker_no	-0.000825	...	-0.001374
smoker_yes	0.000825	...	0.001374
region_northeast	-0.000637	...	0.001560
region_northwest	-0.001005	...	-0.000828
region_southeast	0.000862	...	0.000461
region_southwest	0.000781	...	-0.001193
medical_history_Diabetes	-0.000755	...	0.001048
medical_history_Heart disease	0.001276	...	-0.000536
medical_history_High blood pressure	-0.000128	...	-0.001202
medical_history_None	-0.000393	...	0.000689
family_medical_history_Diabetes	-0.000318	...	0.002248
family_medical_history_Heart disease	-0.000523	...	0.001040
family_medical_history_High blood pressure	0.000609	...	-0.000482
family_medical_history_None	0.000232	...	-0.002804
exercise_frequency_Frequently	-0.000150	...	0.000699
exercise_frequency_Never	0.000901	...	0.000713
exercise_frequency_Occasionally	-0.000162	...	-0.001384
exercise_frequency_Rarely	-0.000588	...	-0.000027
occupation_Blue collar	0.000964	...	0.001658
occupation_Student	-0.000615	...	-0.000818
occupation_Unemployed	-0.000490	...	0.000341
occupation_White collar	0.000142	...	-0.001181
coverage_level_Basic	-0.000046	...	-0.500401
coverage_level_Premium	0.000101	...	-0.499796
coverage_level_Standard	-0.000055	...	1.000000

[31 rows x 31 columns]

Table 4: Correlation matrix of all features in the Insurance dataset (Matplotlib, 2023)

The correlation is visualised with a heat map to demonstrate the relationship among the variables.

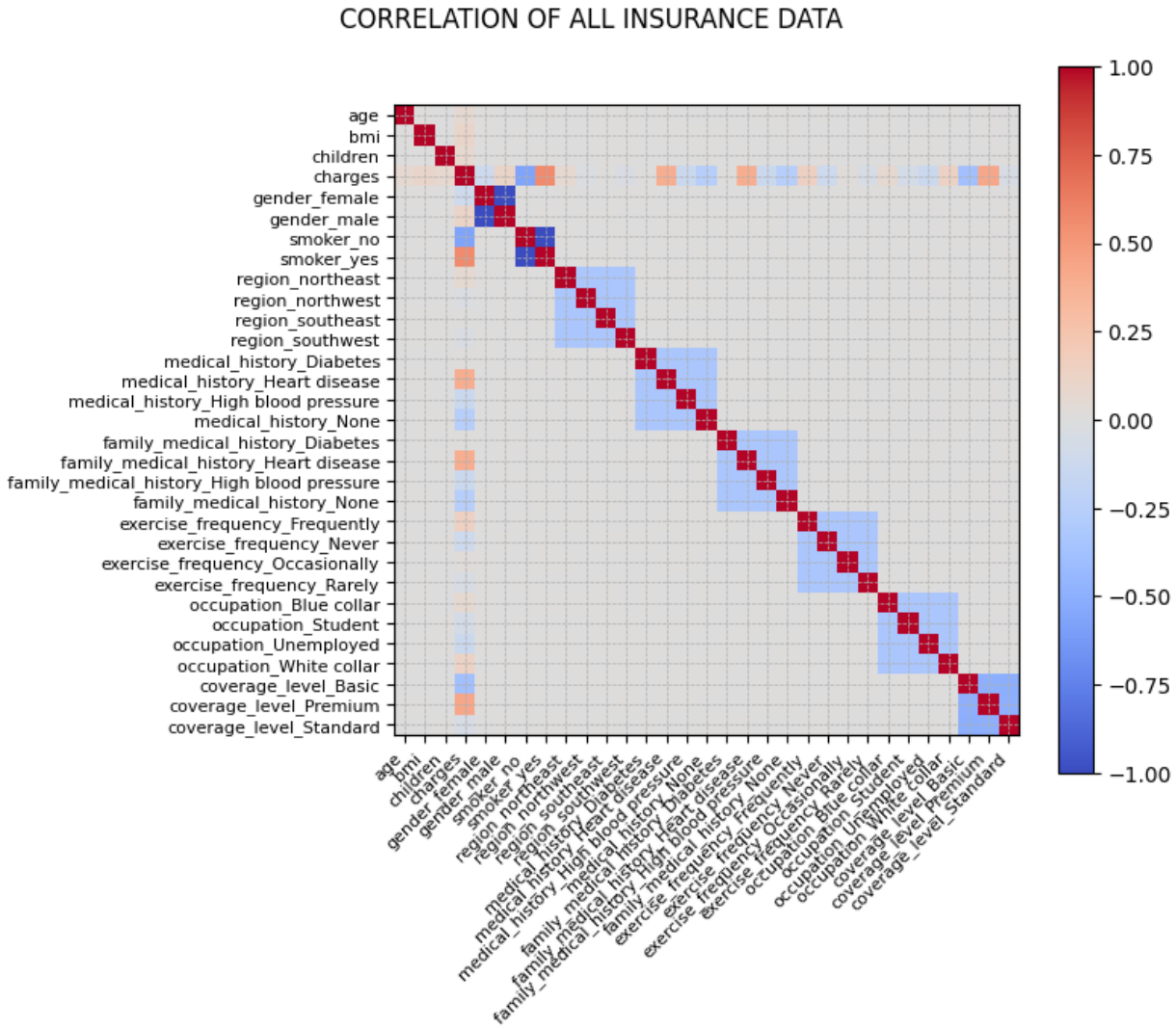

Figure 6: Heatmap of the correlations between all features (Matplotlib, 2023)

It can be observed from the heatmap that Smokers, Personal and Family medical history of Heart disease, Charges and Premium coverage level have high positive correlations, ranging from 0.39 to 0.57. Therefore, policyholders who smoke and have chronic health conditions like heart attacks may tend to purchase the Premium package and pay higher charges for their insurance than others.

On the other hand, charges of non-smokers, people with no personal and family medical history, and Basic coverage level owners are negatively correlated, varying between -0.26 and -0.57. This implies that people with no smoking habit and a healthy medical background pay less for their insurance.

To explain the high correlations, the following graphs are plotted.

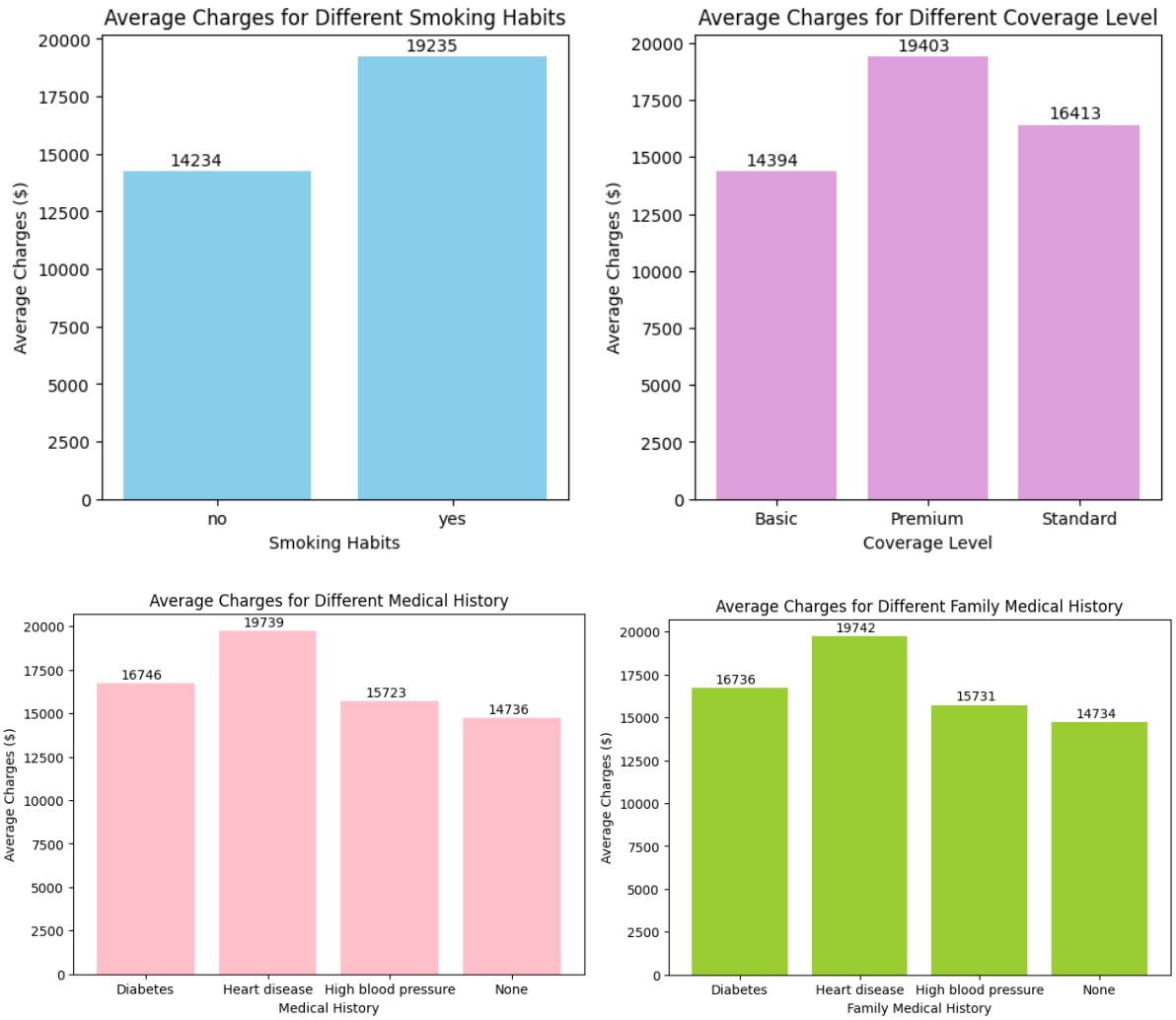

Figure 7: Average charges of policyholders with different smoking habits, medical histories and coverage levels (Matplotlib, 2023)

The bar charts have demonstrated that average charges for policyholders with a smoking habit, heart disease history and Premium coverage ownership are the highest among other factors. Meanwhile, people who do not smoke, have no issues with medical backgrounds and own Basic insurance pay the least average charges. As a result, it can be said that the charges and coverage level of policyholders who smoke and have heart disease risks tend to be higher than those who have healthy habits and other medical conditions.

3.3. The probability of charges from $15,000 for policyholders in the North West, South East, and South West areas is calculated by filtering policyholders from 3 out of 4 regions. From this pool, the number of people with charges higher than $15,000 is divided by the total number of residents of three regions. The following graph depicts the results.

```
{'northwest': 62.04752511693228, 'southeast': 63.94678545533207, 'southwest': 61.27631500029989} %
```

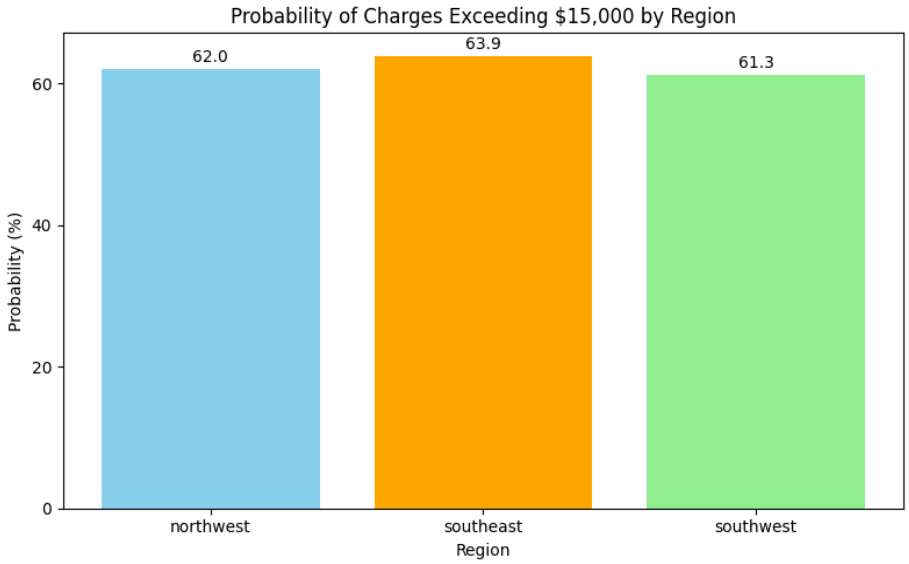

Figure 8: Percentages of charges higher than $15,000 across 3 regions (Matplotlib, 2023)

Among the three regions, the South East accounts for the highest percentage of charges exceeding $15.000 at 63.9%, whereas the South Western part is the lowest at 61.3%.

Deep-diving into other features to analyse this result, the South East is home to policyholders who exercise the least among the three regions. It is also the area with the lowest number of people with no medical history and the highest smoking tendency. These insights prove that people from the South East are less healthy than other areas, which causes them to be charged with higher insurance fees than others.

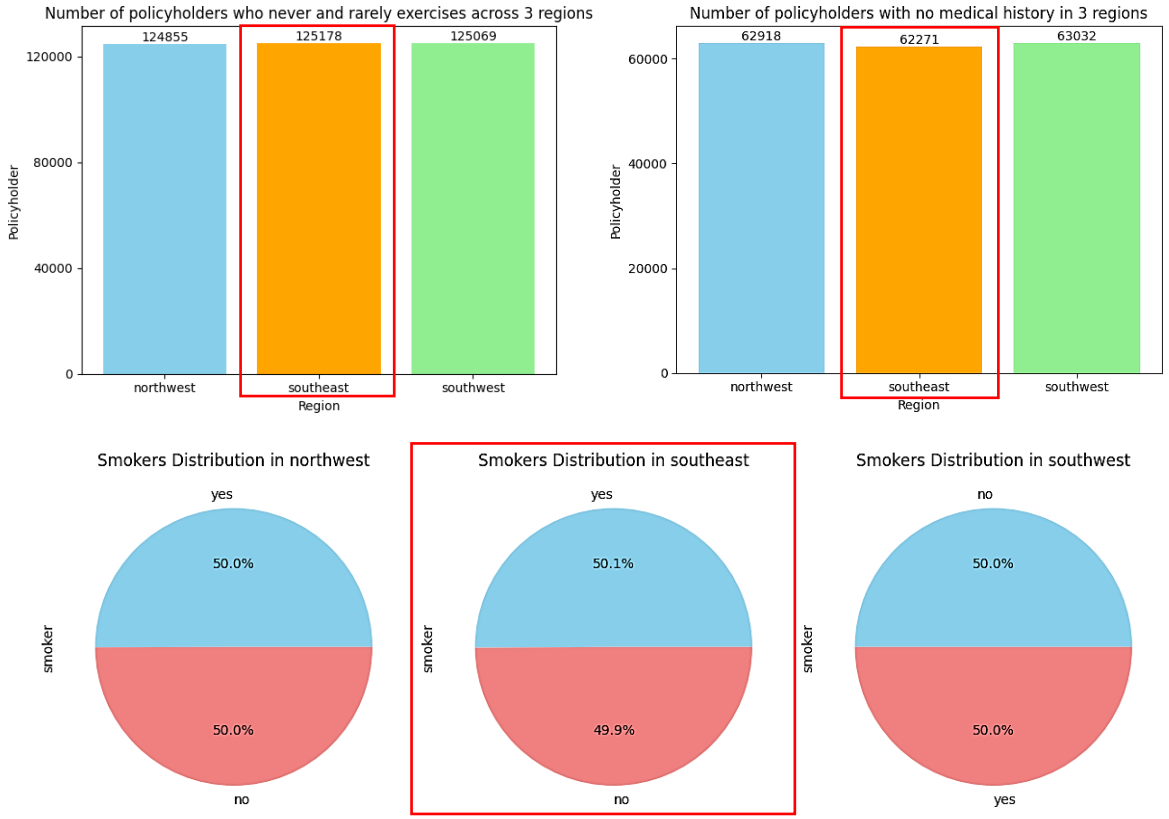

Figure 9: Bar and pie charts on exercise frequency, medical history and smoking status of the North West, South East, and South West regions (Matplotlib, 2023)

3.4. The following bar graphs illustrate the average values of different features:

a. The average BMI for each region is computed and illustrated in the following bar chart. It can be seen that policyholders in all areas share a closely similar BMI index, which is around 34. According to WHO (1995), this metric falls under the category of Obesity class I, which indicates that people from these regions are overweight and might suffer from potential illnesses.

```
region
northeast    34.00
northwest    33.97
southeast    34.02
southwest    34.02
Name: bmi, dtype: float64
```

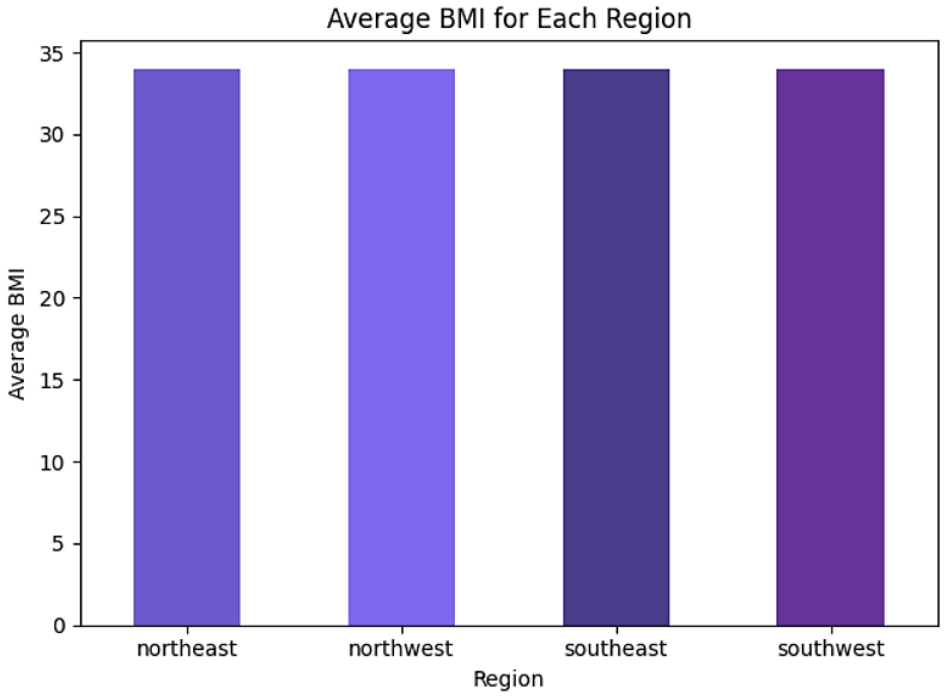

Figure 10: The average BMI of policyholders from 4 regions (Matplotlib, 2023)

b. The graph shows the average charge of the Basic coverage is the lowest at $14,393.92, while the Premium is the highest at $19,402.67. This means that the Premium insurance package which offers more extensive services and benefits requires a higher fee than the Standard and Basic tiers.

```
coverage_level
Basic       14393.92
Premium     19402.67
Standard    16413.06
Name: charges, dtype: float64
```

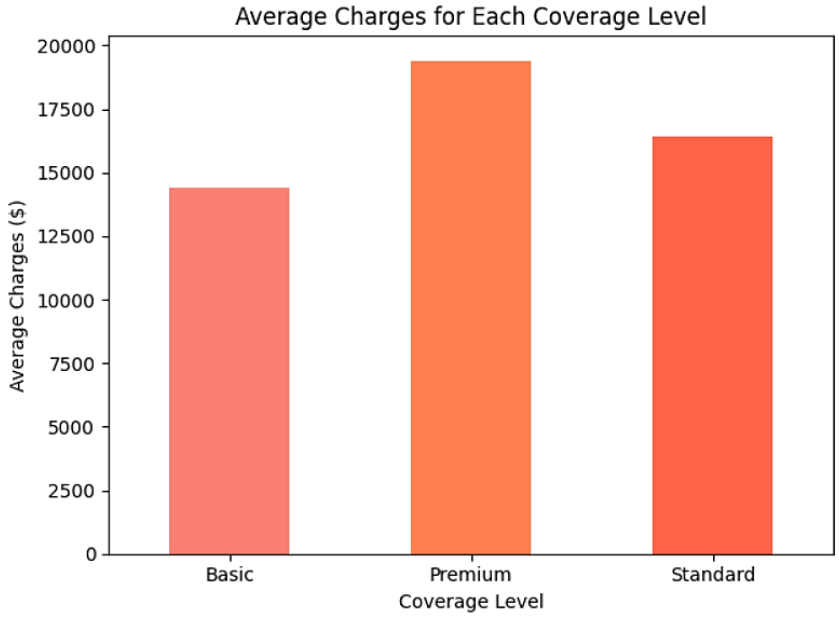

Figure 11: The average charges for 3 coverage levels (Matplotlib, 2023)

c. The average ages of people who smoke and do not are approximately the same, which are 42 and 41 respectively.

```
smoker
no     41.0
yes    42.0
Name: age, dtype: float64
```

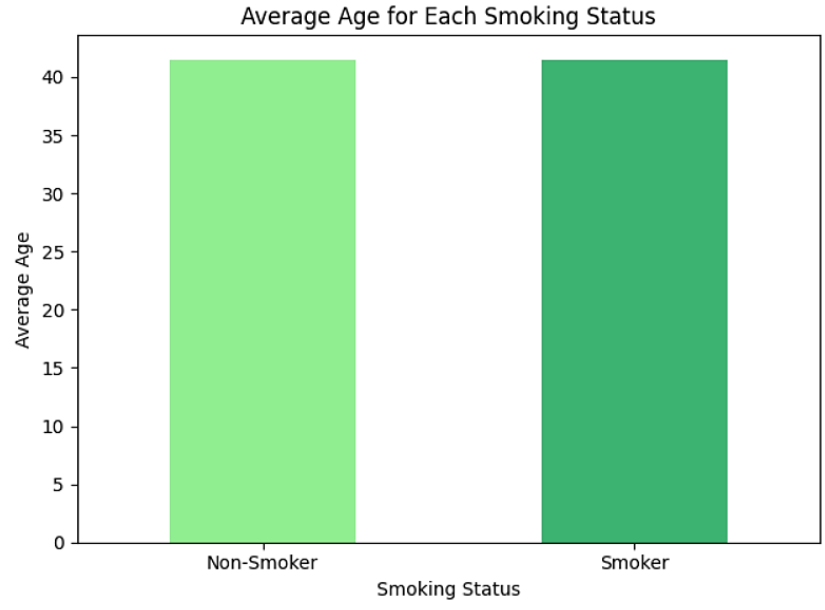

Figure 12: The average age of smokers and non-smokers (Matplotlib, 2023)

IV. TASK 4

4.1. In the scatter plot, an upward regression line indicates a positive relationship between the BMI index and premium charges.

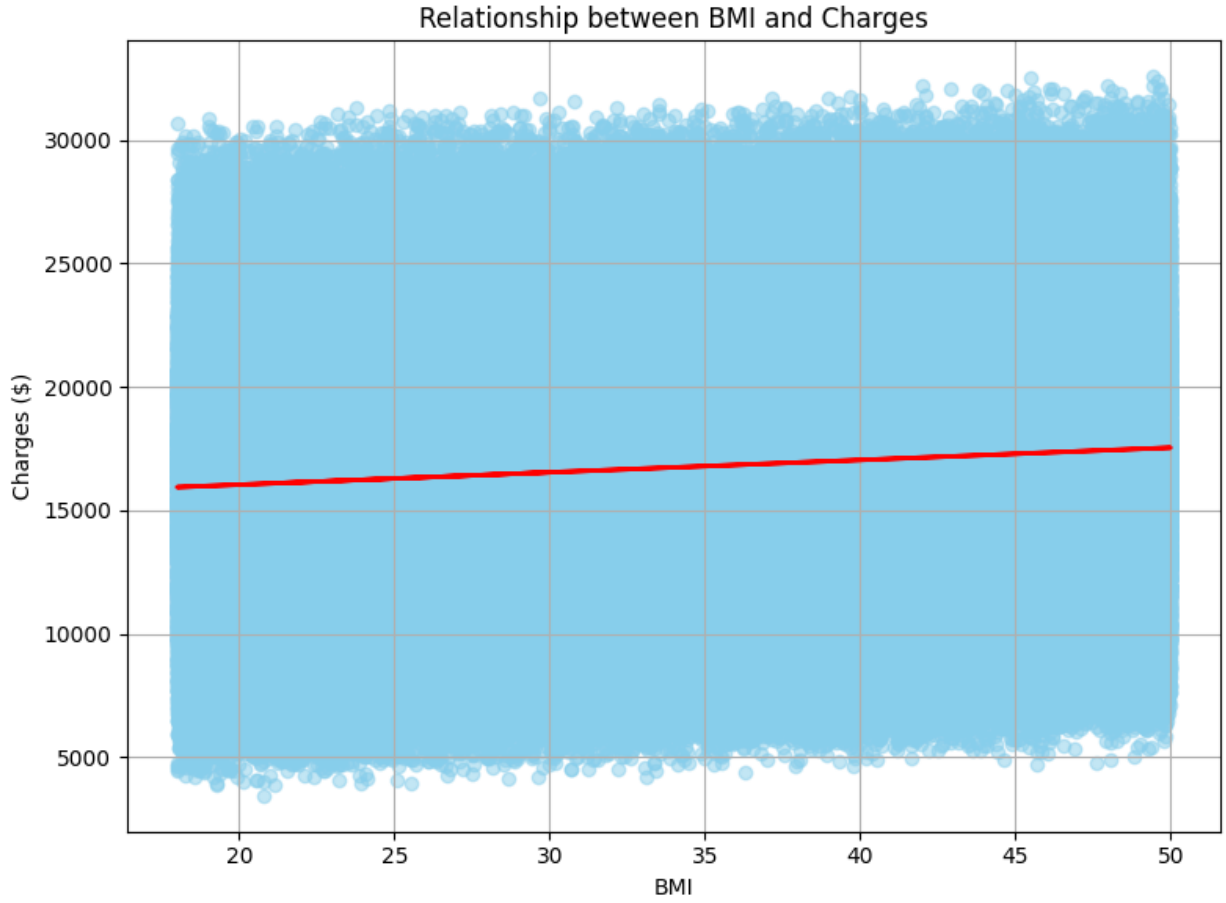

Figure 13: Relationship between BMI index and insurance charges (Matplotlib, 2023)

As the BMI index increases from 18 to 50, the charges also rise. Based on WHO's (1995) categorisation, people with a higher BMI index than 24.9 would be classified as overweight or obese, which are more susceptible to health risks. As a result, these policyholders tend to have higher insurance charges.

4.2. The average fees between males and females are not significantly different, with the former being charged higher than the latter only by 6.2%. It can be said that the gender of a policyholder does not affect one’s cost of insurance.

```
gender
female    16233.70
male      17236.32
Name: charges, dtype: float64
```

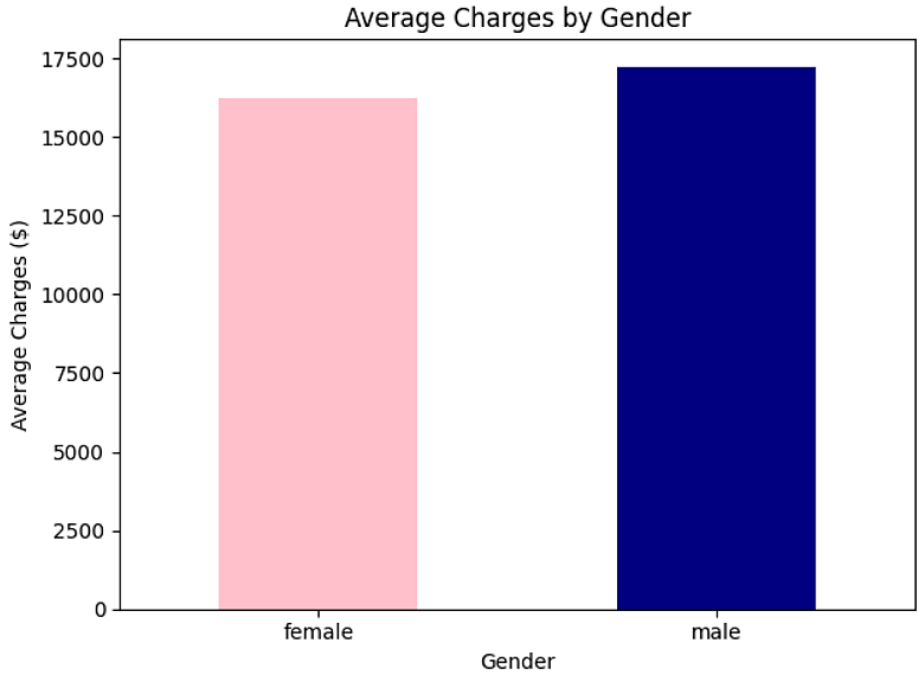

Figure 14: Average charges of male and female policyholders (Matplotlib, 2023)

4.3. The box plot depicts the distribution of the costs for Basic, Standard and Premium insurance levels.

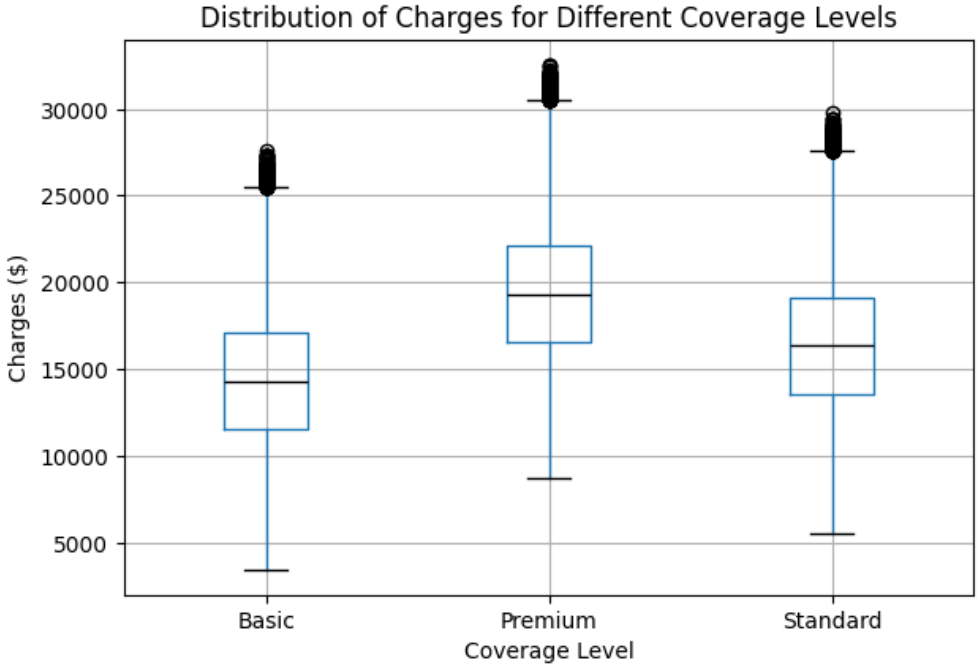

Figure 15: Distribution of charges for 3 coverage levels (Matplotlib, 2023)

The upper and lower whiskers extending from the boxes are the maximum and minimum values of the data. In this case, the fees of the Premium package are the most expensive, ranging from about $9,000 to $31,000, followed by Standard from $6,000 to $27,500, while Basic was

charged the least, from $2,500 to $26,000. Furthermore, all coverage levels have outliers and they lie above the maximum values. This suggests that even though the data within each coverage level is generally distributed around the median, there are a few policyholders with significantly higher charges. The extreme data of the Premium level is the highest, which is $32,500, the Standard is $30,000 and the Basic is the lowest at $27,500.

In conclusion, the box plot shows that the Premium coverage level is the most costly with the highest median and extreme value. On the other hand, the Basic package is the most affordable with the lowest median and smallest outlier charge.

(Word count: 1,346 words)

V. REFERENCES

pandas.DataFrame.corr (2023) Pandas - pandas 2.1.4 documentation. Available at: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

Pavel Tyshevskyi Stack Overflow. (2019). *How encode categorical data without affecting numerical data in a DataFrame?* [online] Available at: https://stackoverflow.com/questions/58969758/how-encode-categorical-data-without-affecting-numerical-data-in-a-dataframe.

WHO Expert Committee (1995). WHO physical status: The use and interpretation of anthropometry: Report of a World Health Organization (WHO) Expert Committee. Geneva.

VI. APPENDIX

The Python programs used for this assignment are PyCharm Community Edition v.2023.2.2 and Jupyter Notebook v.6.5.4. The codes are presented as follows:

TASK 1

```
import pandas as pd
#Import dataset from csv file
directory_path = 'C:/.UoL/Term 1/.Python/Final/insurance_dataset.csv'
df = pd.read_csv(directory_path)
#Display the first 8 rows of dataframe
print(df.head(8))

#Count the shape of the dataset
print(df.shape)
print(f'The insurance data-set contains {df.shape[0]} rows and {df.shape[1]} columns')
```

TASK 2

```
#Check number of missing values
print(df.isna().sum())

#Descriptive statistics of numeric columns
print(df.describe())

#Descriptive statistics of categorical columns
print(df.describe(include=["object","bool"]))
```

TASK 3

3.1.

```
import matplotlib.pyplot as plt

#3.1. Plot distribution of policyholders across various regions
##Using a bar plot
plt.figure(figsize=(12, 6))
region_counts = df['region'].value_counts()
colors = ['blue', 'lightskyblue', 'cornflowerblue', 'deepskyblue']
plt.bar(region_counts.index, region_counts, color=colors)
plt.title('Distribution of Policyholders Across Regions')
plt.xlabel('Region')
plt.ylabel('Number of Policyholders')
plt.tight_layout()
plt.show()

##Using a pie chart
plt.figure(figsize=(6,6))
colors = ['bisque', 'tomato', 'sandybrown', 'orange']
pie = plt.pie(region_counts, labels=region_counts.index, autopct='%.2f%%', startangle=90, colors=colors)
for text in pie[2]:
    text.set_bbox(dict(facecolor='white'))
plt.title('Distribution of Policyholders Across Regions')
plt.tight_layout()
plt.show()
```

```
##Plot a heatmap
###Create a heatmap
region_counts = df['region'].value_counts().reset_index(name='Policyholders')
regions = region_counts['index'].tolist()
policyholders = region_counts['Policyholders'].tolist()
policyholders.reverse()
fig, ax = plt.subplots(figsize=(10, 6))
heatmap = ax.imshow([policyholders], cmap="GnBu", aspect='auto', extent=[len(regions) + 0.5, 0.5, 0, 1])

###Add colorbar
cbar = fig.colorbar(heatmap, ax=ax, orientation='horizontal', pad=0.1, shrink=1)
cbar.set_label('Policyholders', labelpad=10)

###Customise axes and labels
ax.set_xticks(np.arange(1, len(regions) + 1))
ax.set_xticklabels(regions)
ax.set_yticks([])
plt.title('Distribution of Policyholders Across Regions')
plt.xlabel('Region')
plt.tight_layout()
plt.show()
```

3.2.

```
# Create dummy variables for categorical columns
df = pd.get_dummies(df, columns=['gender', 'smoker', 'region', 'medical_history', 'family_medical_history', 'exercise_frequency', 'occupation', 'coverage_level'])

# Calculate correlations
correlations = df.corr()
print(correlations)
# Visualise the correlation matrix with a heatmap
fig, ax = plt.subplots(figsize=(8, 7))
plt.imshow(correlations, cmap='coolwarm', interpolation='nearest')
plt.colorbar()
plt.xticks(range(len(correlations.columns)), correlations.columns, rotation=45, ha='right', fontsize=8)
plt.yticks(range(len(correlations.columns)), correlations.columns, fontsize=8)
plt.suptitle('CORRELATION OF ALL INSURANCE DATA')
plt.tight_layout()
plt.show()
```

```
#Filter Smoker, Medical history, Family medical history and Coverage features
def plot_feature_bar_chart(feature, title, color):
    grouped_data = df.groupby([feature])['charges'].mean().reset_index()

# Plot bar charts for each feature
    plt.figure(figsize=(7, 5))
    bars = plt.bar(grouped_data[feature], grouped_data['charges'], color=color)
    for bar, value in zip(bars, grouped_data['charges']):
        plt.text(bar.get_x() + bar.get_width() / 2 - 0.2, value + 255, f'{round(value)}', ha='left', color='black')
    plt.title(f'Average Charges for Different {title}')
    plt.xlabel(f'{title}')
    plt.ylabel('Average Charges ($)')
plot_feature_bar_chart('smoker', 'Smoking Habits', color='skyblue')
plot_feature_bar_chart('medical_history', 'Medical History', color='pink')
plot_feature_bar_chart('family_medical_history', 'Family Medical History', color='yellowgreen')
plot_feature_bar_chart('coverage_level', 'Coverage Level', color='plum')
plt.tight_layout()
plt.show()
```

3.3.

```
#Filter the data for policyholders from 3 regions
regions = ["northwest", "southeast", "southwest"]
filtered_data = df[df['region'].isin(regions)]

#Calculate the probability of charges >$15,000 for each region
probabilities = {}
for region in regions:
    region_data = filtered_data[filtered_data['region'] == region]
    total_count = len(region_data)
    exceeding_count = len(region_data[region_data['charges'] > 15000])
    probability = exceeding_count / total_count*100
    probabilities[region] = probability
print(probabilities,"%")

#Visualise with a bar chart
plt.bar(probabilities.keys(), probabilities.values(), color=['skyblue', 'orange', 'lightgreen'])
plt.xlabel("Region")
plt.ylabel("Probability (%)")
plt.yticks([0, 20, 40, 60])
plt.title("Probability of Charges Exceeding $15,000 for Different Regions")
for i, v in enumerate(probabilities.values()):
    plt.text(i, v + 1, str(round(v, 1)), ha='center')
plt.tight_layout()
plt.show()
```

```
#Compare smokers in each region
def compare_smokers(region, ax):
    subset = filtered_data[filtered_data['region'] == region]
    subset['smoker'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=['skyblue', 'lightcoral'], ax=ax)
    ax.set_title('Smokers Distribution in {}'.format(region))

# Display pie charts
fig, axs = plt.subplots(1, 3, figsize=(12, 4))
for i, region in enumerate(regions):
    compare_smokers(region, axs[i])
plt.tight_layout()
plt.show()
```

```
#Filter the people with no Medical History for each region
none_counts = filtered_data[filtered_data['medical_history'] == 'None']['region'].value_counts()

#Visualise result with bar chart
plt.bar(regions, none_counts[regions], color=['skyblue', 'orange', 'lightgreen'])
for i, v in enumerate(none_counts[regions].values):
    plt.text(i, v + 0.1, str(v), ha='center', va='bottom')
plt.title('Number of policyholders with no medical history in 3 regions')
plt.xlabel('Region')
plt.ylabel('Policyholder')
plt.yticks([0, 20000, 40000, 60000])
plt.tight_layout()
plt.show()
```

```
#Filter the number of people who never or rarely exercises for each region
exercise_counts = filtered_data[filtered_data['exercise_frequency'].isin(['Never', 'Rarely'])]['region'].value_counts()

#Visualise result with bar chart
plt.bar(regions, exercise_counts[regions], color=['skyblue', 'orange', 'lightgreen'])
for i, v in enumerate(exercise_counts[regions].values):
    plt.text(i, v + 0.1, str(v), ha='center', va='bottom')
plt.title('Number of policyholders who never and rarely exercises across 3 regions')
plt.xlabel('Region')
plt.ylabel('Policyholder')
plt.yticks([0, 40000, 80000, 120000])
plt.tight_layout()
plt.show()
```

3.4.

```
# (a) Calculate average BMI for each region
average_bmi_by_region = df.groupby('region')['bmi'].mean()
print('(a) Average BMI for each region:', round(average_bmi_by_region,2))

#Plot the bar chart
average_bmi_by_region.plot(kind='bar', color=['slateblue', 'mediumslateblue', 'darkslateblue', 'rebeccapurple'])
plt.title('Average BMI for Each Region')
plt.xlabel('Region')
plt.xticks(rotation=0)
plt.ylabel('Average BMI')
plt.tight_layout()
plt.show()
```

```
# (b) Calculate average charges for each coverage level
average_charges_by_coverage = df.groupby('coverage_level')['charges'].mean()
print('(b) Average charges for each coverage level:', round(average_charges_by_coverage,2))

#Plot the bar chart
average_charges_by_coverage.plot(kind='bar', color=['salmon','coral','tomato'])
plt.title('Average Charges for Each Coverage Level')
plt.xlabel('Coverage Level')
plt.xticks(rotation=0)
plt.ylabel('Average Charges ($)')
plt.tight_layout()
plt.show()
```

```
# (c) Calculate average age for smokers and non-smokers
average_age_by_smoking_status = df.groupby('smoker')['age'].mean()
average_age_by_smoking_status.plot(kind='bar', color=['lightgreen','mediumseagreen'])
print('(c) Average age for each smoking status:', round(average_age_by_smoking_status,0))

#Plot the bar chart
plt.title('Average Age for Each Smoking Status')
plt.xlabel('Smoking Status')
plt.ylabel('Average Age')
plt.xticks([0, 1], ['Non-Smoker', 'Smoker'], rotation=0)
plt.tight_layout()
plt.show()
```

TASK 4

4.1.

```
#Filter BMI and Charges data
bmi = df['bmi'].values.reshape(-1, 1)
charges = df['charges']

#Draw a linear regression
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(bmi, charges)
predicted_charges = model.predict(bmi)

#Plot the scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(bmi, charges, alpha=0.5, color='skyblue')
plt.plot(bmi, predicted_charges, color='red', linewidth=2, label='Regression Line')
plt.title('Relationship between BMI and Charges')
plt.xlabel('BMI')
plt.ylabel('Charges ($)')
plt.grid(True)
plt.tight_layout()
plt.show()
```

4.2.

```
#Calculate the average charges between males and females
df_gender = df.groupby('gender')['charges'].mean()
print(round(df_gender, 2))

#Plot a bar graph
df_gender.plot(kind='bar', color=['pink','navy'], xlabel='Gender', ylabel='Average Charges ($)', title='Average Charges by Gender')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

4.3.

```
# Illustrate distribution of insurance charges across coverage levels with a Box plot
plt.figure(figsize=(8, 6))
df.boxplot(column='charges', by='coverage_level', grid=True, medianprops={'color':'black'})
plt.title('Distribution of Charges for Different Coverage Levels')
plt.xlabel('Coverage Level')
plt.ylabel('Charges ($)')
plt.tight_layout()
plt.show()
```