U.S. Medical Insurance Costs

Codecademy provides students with the U.S. Medical Insurance Costs dataset to build a portfolio.

In [1]:
```
import pandas as pd

# Load the data from the insurance.csv file into a DataFrame
data = pd.read_csv('insurance.csv')

#Display all data
data
```

Out[1]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [2]:
```
# Display the first few rows of the data
data.head()
```

Out[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Questions from Coursera:

1. Find out the average age of the patients in the dataset.

In [3]:
```
# Calculate the average age of the patients
average_age = data['age'].mean()
average_age
```

Out[3]: 39.20702541106129

In [4]:
```
# Calculate the average age of the patients and round to the nearest whole number
average_age = round(data['age'].mean())
average_age
```

Out[4]: 39

Answer: 39

2. Analyse where a majority of the individuals are from.

In [5]:
```
# Count the number of individuals from each region
region_counts = data['region'].value_counts()
region_counts
```

Out[5]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

Answer: Southeast with a total of 364

3. Look at the different costs between smokers vs. non-smokers.

In [6]:
```
# Compare the charges for smokers vs. non-smokers
smoker_charges = data.groupby('smoker')['charges'].mean()
smoker_charges
```

Out[6]:
```
smoker
no      8434.268298
yes    32050.231832
Name: charges, dtype: float64
```

In [7]:
```
# Compare the charges for smokers vs. non-smokers and round to the nearest whole number
smoker_charges = round(data.groupby('smoker')['charges'].mean())
smoker_charges
```

Out[7]:
```
smoker
no      8434.0
yes    32050.0
Name: charges, dtype: float64
```

Answer:: Non-Smokers get charged 8432 USD and Smokers get charged 32050 USD on average.

4. Figure out what the average age is for someone who has at least one child in this data set.

In [8]:
```
# Calculate the average age of individuals who have at least one child
average_age_with_child = data[data['children'] >= 1]['age'].mean()
average_age_with_child
```

Out[8]: 39.78010471204188

In [9]:
```
# Calculate the average age of individuals who have at least one child and round to the
average_age_with_child = round(data[data['children'] >= 1]['age'].mean())
average_age_with_child
```

Out[9]: 40

Answer: 40

Extra Analysis by myself

Using a regression analysis and predictive models to determine factors that influence insurance charges

In [10]:
```
# Generate descriptive statistics for the different variables
data.describe(include='all')
```

Out[10]:

	age	sex	bmi	children	smoker	region	charges
count	1338.000000	1338	1338.000000	1338.000000	1338	1338	1338.000000
unique	NaN	2	NaN	NaN	2	4	NaN
top	NaN	male	NaN	NaN	no	southeast	NaN
freq	NaN	676	NaN	NaN	1064	364	NaN
mean	39.207025	NaN	30.663397	1.094918	NaN	NaN	13270.422265
std	14.049960	NaN	6.098187	1.205493	NaN	NaN	12110.011237
min	18.000000	NaN	15.960000	0.000000	NaN	NaN	1121.873900
25%	27.000000	NaN	26.296250	0.000000	NaN	NaN	4740.287150
50%	39.000000	NaN	30.400000	1.000000	NaN	NaN	9382.033000
75%	51.000000	NaN	34.693750	2.000000	NaN	NaN	16639.912515
max	64.000000	NaN	53.130000	5.000000	NaN	NaN	63770.428010

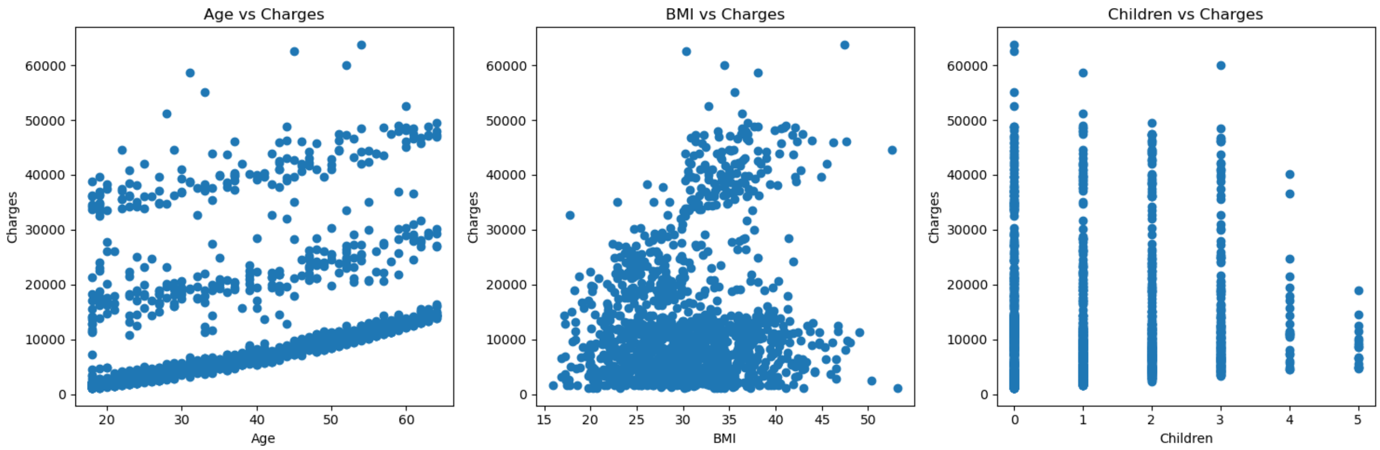
In [11]:
```
# Calculate the correlation between the different variables and the insurance charges
correlations = data.corr()['charges'].sort_values()
correlations
```

```
C:\Users\jonat\AppData\Local\Temp\ipykernel_78392\1034574442.py:2: FutureWarning: The de
fault value of numeric_only in DataFrame.corr is deprecated. In a future version, it wil
l default to False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  correlations = data.corr()['charges'].sort_values()
```

Out[11]:
```
children    0.067998
bmi         0.198341
age         0.299008
charges     1.000000
Name: charges, dtype: float64
```

In [12]:
```
import matplotlib.pyplot as plt

# Create scatter plots for the numerical variables
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.scatter(data['age'], data['charges'])
plt.title('Age vs Charges')
plt.xlabel('Age')
plt.ylabel('Charges')

plt.subplot(1, 3, 2)
plt.scatter(data['bmi'], data['charges'])
```

```python
plt.title('BMI vs Charges')
plt.xlabel('BMI')
plt.ylabel('Charges')

plt.subplot(1, 3, 3)
plt.scatter(data['children'], data['charges'])
plt.title('Children vs Charges')
plt.xlabel('Children')
plt.ylabel('Charges')

plt.tight_layout()
plt.show()
```

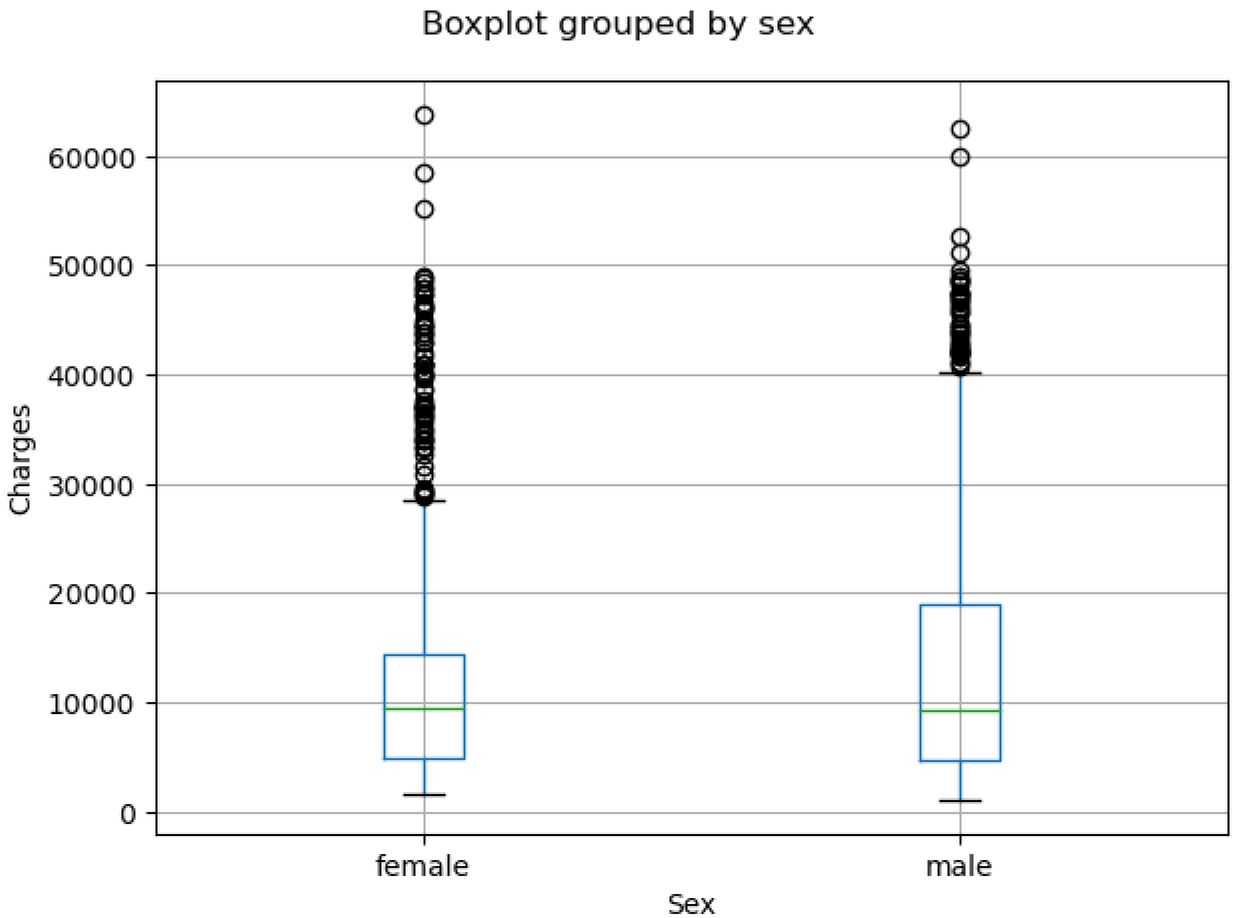


Age vs Charges: There seems to be a positive correlation betwen age and charges, which aligns with our correlation analysis. As age increases, the insurance charges also tend to increase.

BMI vs Charges: There is a less clear pattern between BMI and Charges, but it seems that higher charges are common among individuals with highe BMI.

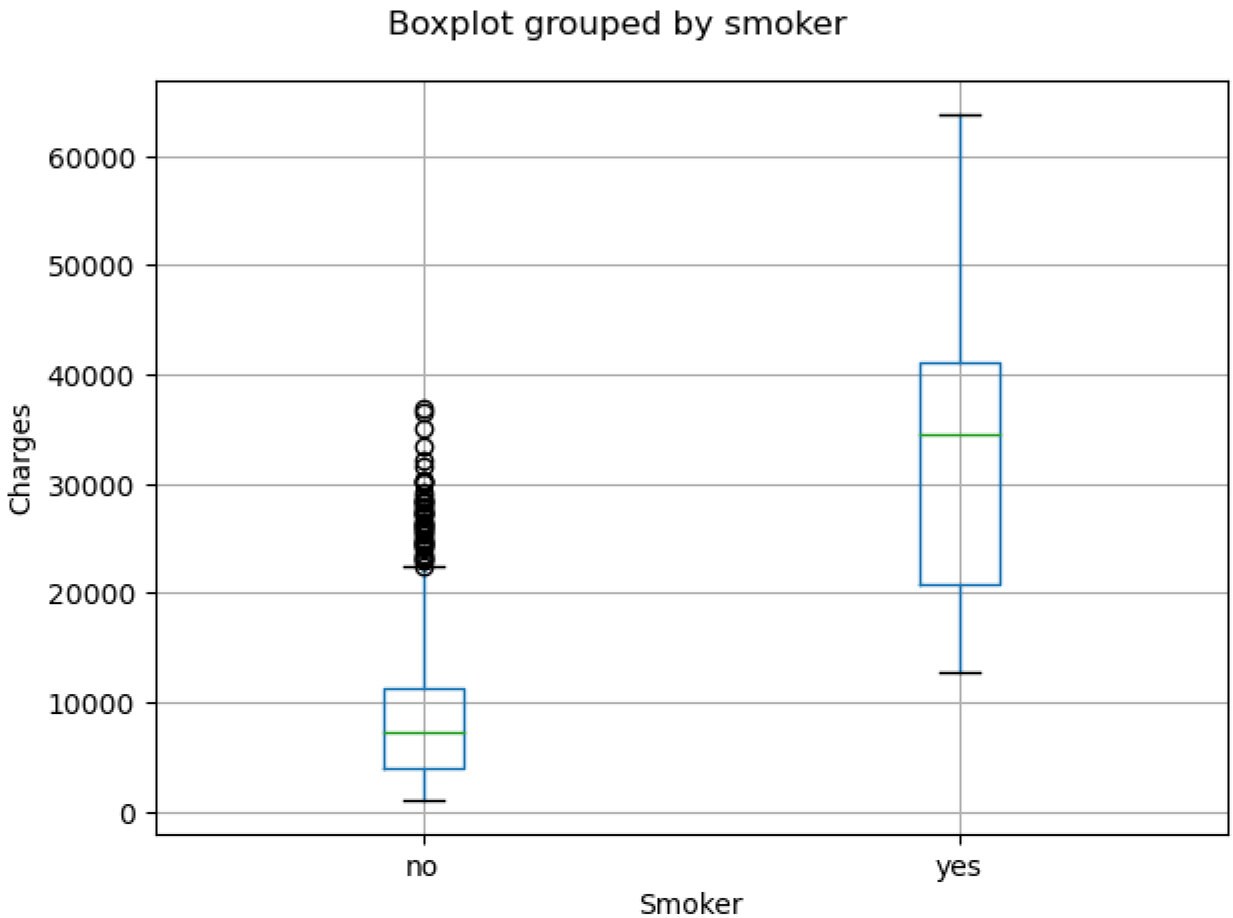
Children vs Charges: There doesn't seem to be a strong relationship between the number of children and charges.

In [13]:
```python
# Create box plots for the categorical variables
data.boxplot(column='charges', by='sex')
plt.title('')
plt.xlabel('Sex')
plt.ylabel('Charges')

plt.tight_layout()
plt.show()
```



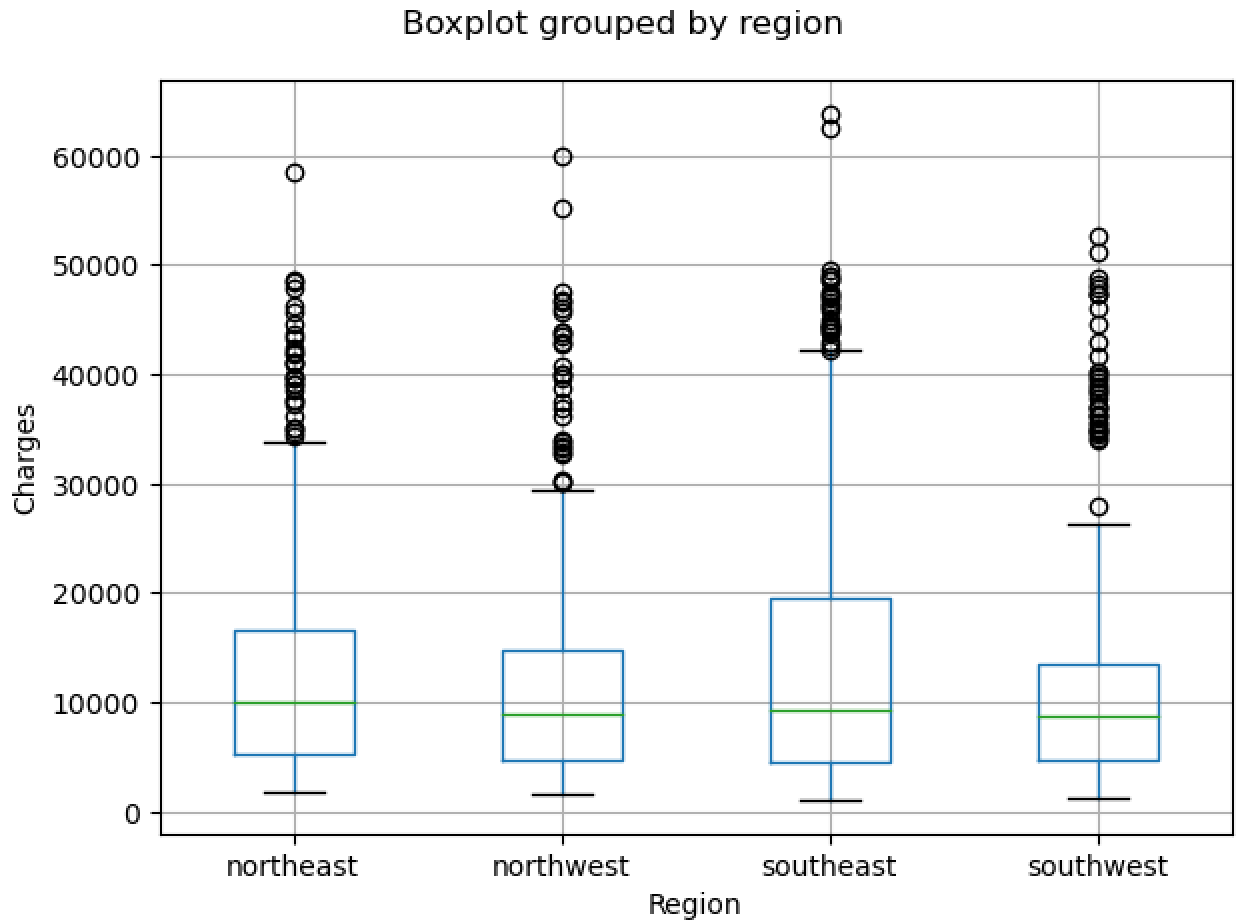
In [14]:

```
data.boxplot(column='charges', by='smoker')
plt.title('')
plt.xlabel('Smoker')
plt.ylabel('Charges')

plt.tight_layout()
plt.show()
```



In [15]:

```
data.boxplot(column='charges', by='region')
plt.title('')
plt.xlabel('Region')
plt.ylabel('Charges')

plt.tight_layout()
plt.show()
```



Sex vs Charges: There doesn't seem to be a significant difference in charges between males and females.

Smoker vs charges: There is a clear difference in charges between smokers and non-smokers. Smokers tend to have much higher charges.

Region vs Charges: There doesn't seem to be a significant difference in charges between the different region.

Regression Models

I build multiple regression models to quantify the relationships between the variables and the insurance charges. This will allow us to identify the most significant factors. However, before we can do this, we need to convert the categorical variables into numerical form.

Below is a new code cell to the notebook to perform this conversion.

In [16]:
```
from sklearn.preprocessing import LabelEncoder

# Create a copy of the data
data_encoded = data.copy()

# Create a label encoder
le = LabelEncoder()

# Convert the categorical variables into numerical form
data_encoded['sex'] = le.fit_transform(data_encoded['sex'])
data_encoded['smoker'] = le.fit_transform(data_encoded['smoker'])
data_encoded['region'] = le.fit_transform(data_encoded['region'])
```

```
data_encoded.head()
```

Out[16]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

Above are the first few rows of the data after conversion

In [17]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Define the feature variables and the target variable
X = data_encoded.drop('charges', axis=1)
y = data_encoded['charges']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

# Create a linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the mean squared error and the R-squared value
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

mse, r2
```

Out[17]:

```
(33635210.43117841, 0.7833463107364538)
```

Here are the performance metrics for the model on the test set:

Mean Squared Error (MSE): 33635210.43 R-squared (R^2): 0.7833

The R-squared value indicates that approximately 78.33% of the variability in the insurance charges can be explained by the variables in our model. This suggests that the model has a relatively good fit to the data.

Examining the coefficients of the model to identify the most significant factors.

In [18]:

```
# Get the coefficients of the model
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
coefficients
```

Out[18]:

	Coefficient
age	257.056264
sex	-18.791457

bmi	335.781491
children	425.091456
smoker	23647.818096
region	-271.284266

The coefficients represent the change in the insurance charges for a one-unit increase in the corresponding variable, assuming all other variables are held constant.

From the coefficients, we can see that smoker has the largest coefficient, indicating that being a smoker has the most significant impact on insurance charges. This is followed by children, bmi, and age. The sex and region variables have relatively small coefficients, suggesting that they have less impact on insurance charges.