Getting the Data

ACME Insurance has a Dataset with verified historical data consisting of information and medical charges incurred by over 1300 customers.

This Dataset will be used to create a Linear Regression Model

Importing Libraries

Lets import the necessary libraries

```
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import SGDRegressor
%matplotlib inline
```

Retrieving the Data

Lets take the data and prepare it for use

We can check the number of rows and columns by calling the dataframe

```
medical_df = pd.read_csv("insurance.csv")
medical_df
```

	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 x 7 columns]

The dataset has 1338 rows(variables) and 7 columns(features). Each row contains the information for one customer

The objective of our model is to determine the value in the "charges" column based on the values in the other columns. If we can determine the historical values, we can estimate the charges for new customers too using the information in the other columns.

Lets check the data types of each column.

```
medical_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

The "Age", "Children", "BMI (body mass index)" and "Charges" columns have number values

"Sex", "Smoker" and "Region" are strings.

We do not have any null values in our columns.

Lets get the summary statistics (Mean, IQR, Max, Min, STD, Count) for our dataset using the .describe function

```
medical_df.describe()
```

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

Exploratory Data Analysis and Visualization

Lets do some EDA by visualizing the distribution of values in certain columns and relationship between the "Charges" and other columns

The following settings will improve the default style and font sizes for our charts

```
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (10, 6)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

Age

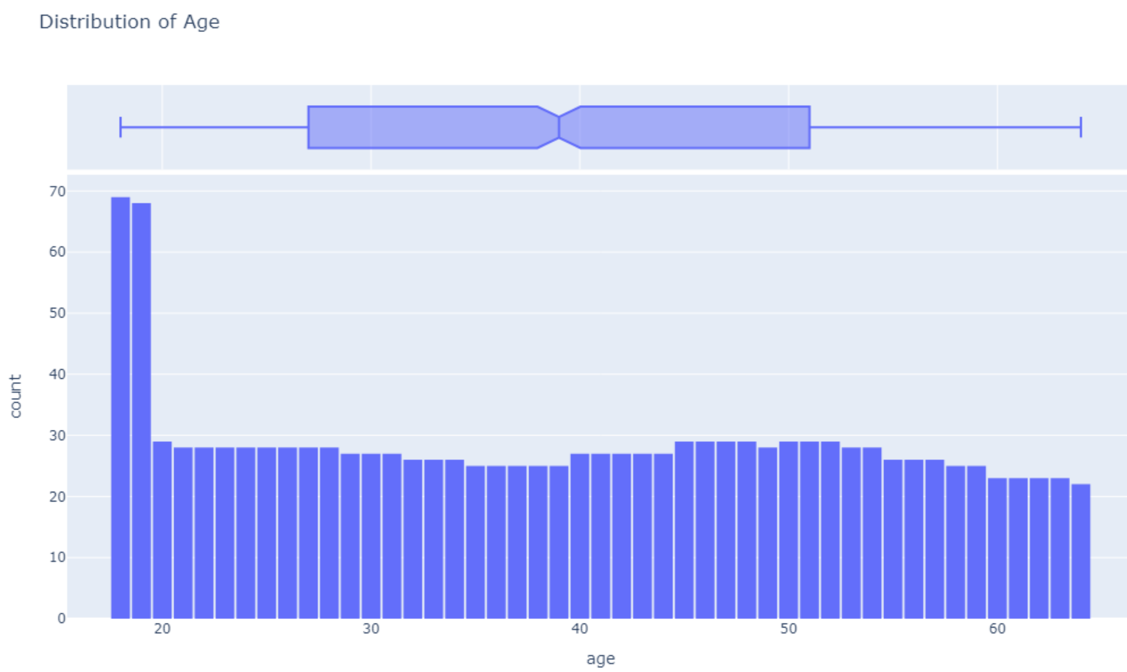
Age is a numeric column. The min age is 18 and max is 64. We can visualize the distribution using a histogram and a box plot.

We'll use plotly to make the chart interactive

```
medical_df.age.describe()
```

```
count    1338.000000
mean       39.207025
std        14.049960
min        18.000000
25%        27.000000
50%        39.000000
75%        51.000000
max        64.000000
Name: age, dtype: float64
```

```
fig = px.histogram(medical_df,
                   x='age',
                   marginal = 'box',
                   nbins = 47,
                   title = 'Distribution of Age')
fig.update_layout(width=800, height=650, bargap=0.1)
fig.show()
```



From the histogram, we notice that the distribution of the ages is almost uniform with about 20-30 customers at every age except for ages 18 and 19

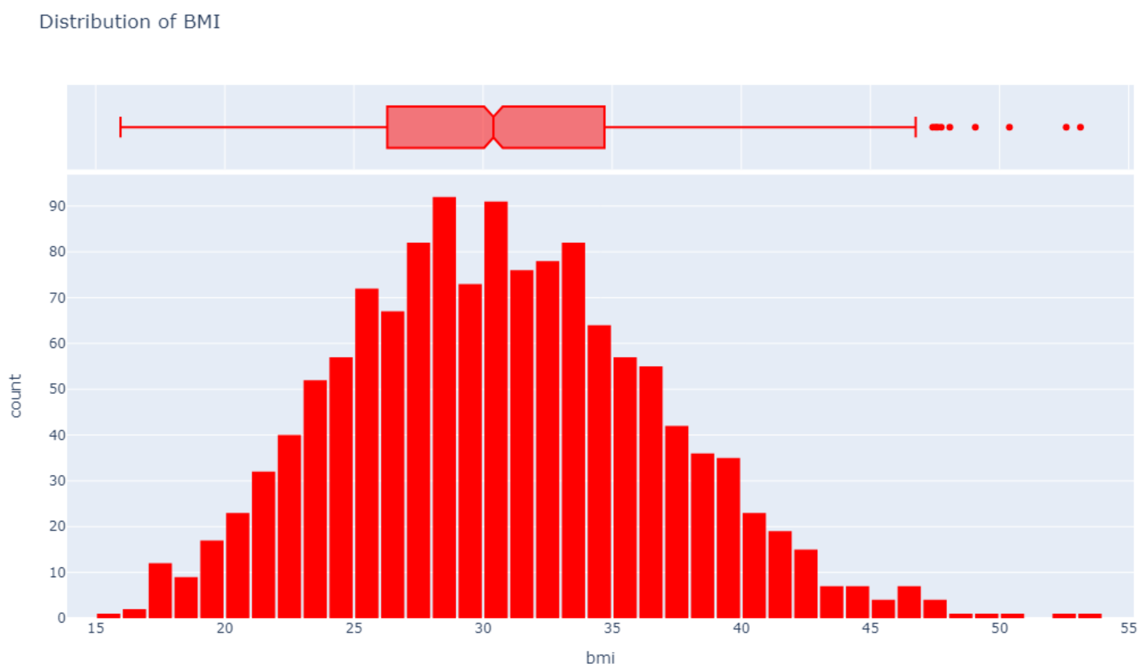
Body Mass Index

Lets look at the distribution of the BMI of customers.

```
medical_df.bmi.describe()
```

```
count    1338.000000
mean       30.663397
std         6.098187
min        15.960000
25%        26.296250
50%        30.400000
75%        34.693750
max        53.130000
Name: bmi, dtype: float64
```

```
fig = px.histogram(medical_df,
                   x='bmi',
                   marginal = 'box',
                   color_discrete_sequence = ['red'],
                   title = 'Distribution of BMI')
fig.update_layout(width=800, height=650, bargap=0.1)
fig.show()
```



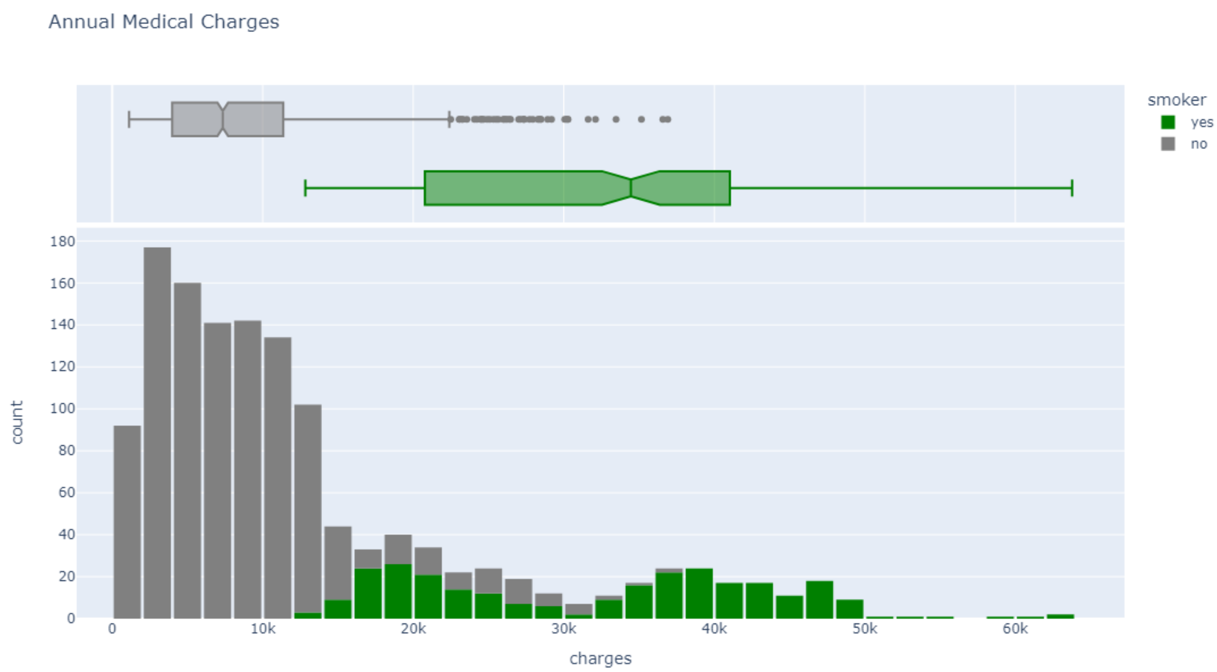
The BMI Measurements form a Gaussian Distribution around the value 30

Charges

Lets look at the distribution of the charges column which we are trying to predict.

We can distinguish between smoker and non-smoker using different colors

```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal = 'box',
                   color = 'smoker',
                   color_discrete_sequence = ['green', 'grey'],
                   title= 'Annual Medical Charges')
fig.update_layout(width=800, height=650, bargap=0.1)
fig.show()
```



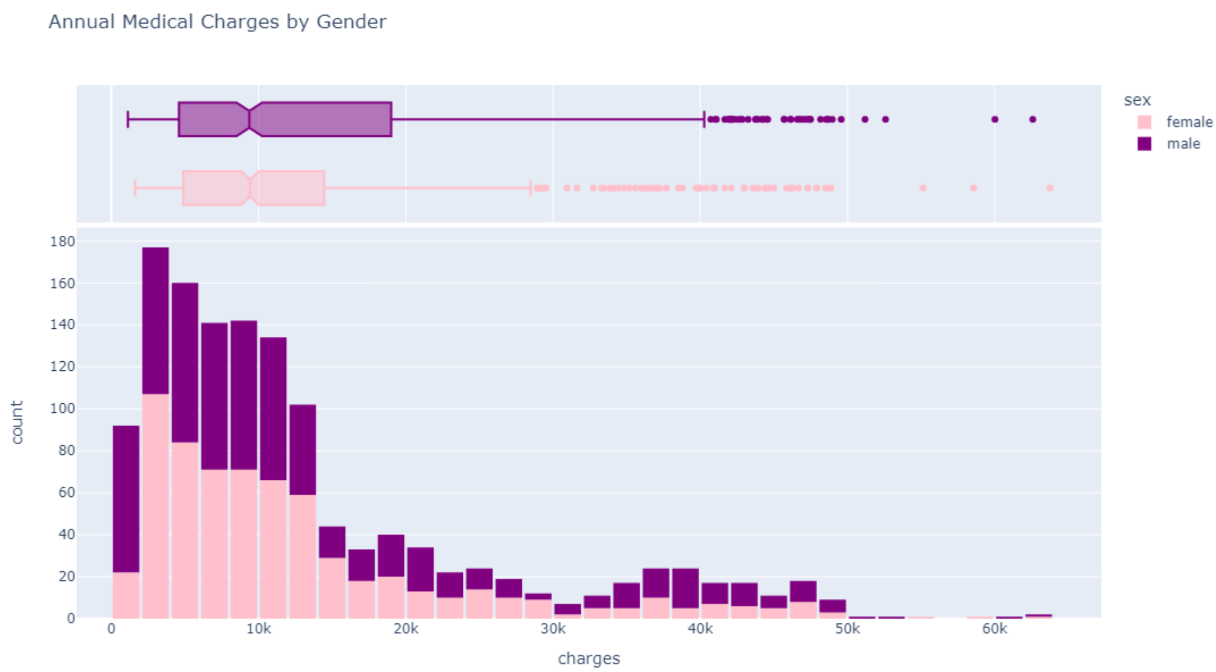
Generally, more people spend less than 10000 on annual medical charges.

It seems that people who smoke have higher median charges

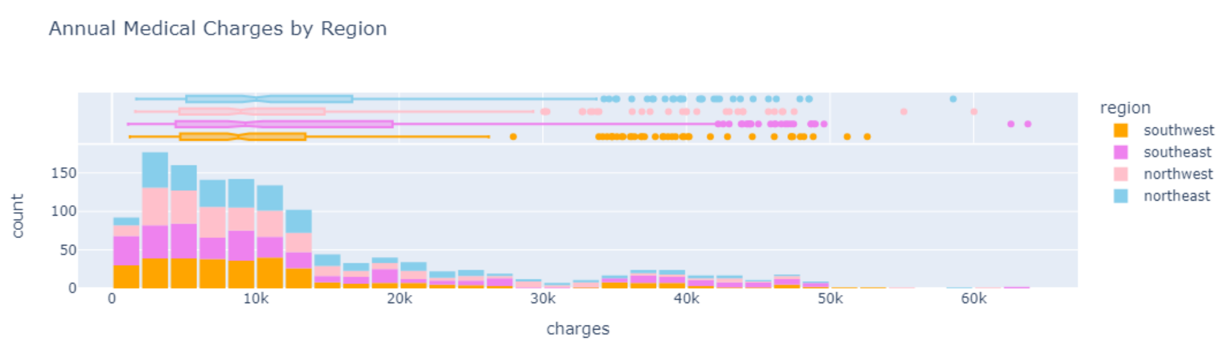
Relationships

Lets visualize the relationship between the "Charges" and other factors like "sex" etc.

```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal = 'box',
                   color = 'sex',
                   color_discrete_sequence = ['pink', 'purple'],
                   title= 'Annual Medical Charges by Gender')
fig.update_layout(width=800, height=650, bargap=0.1)
fig.show()
```




```
fig = px.histogram(medical_df,
                   x='charges',
                   marginal = 'box',
                   color = 'region',
                   color_discrete_sequence = ['orange', 'violet',
'pink', 'skyblue'],
                   title= 'Annual Medical Charges by Region')
fig.update_layout(width=700, height=350, bargap=0.1)
fig.show()
```



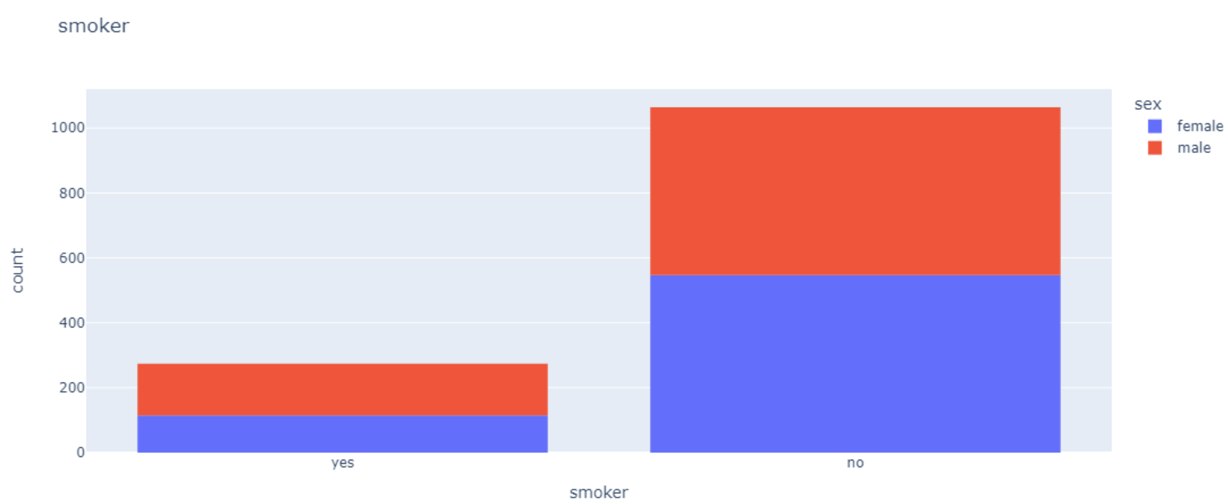

```
medical_df.smoker.value_counts()

smoker
no      1064
```

```
yes      274
Name: count, dtype: int64
```

We can Visualize the distribution of every single column

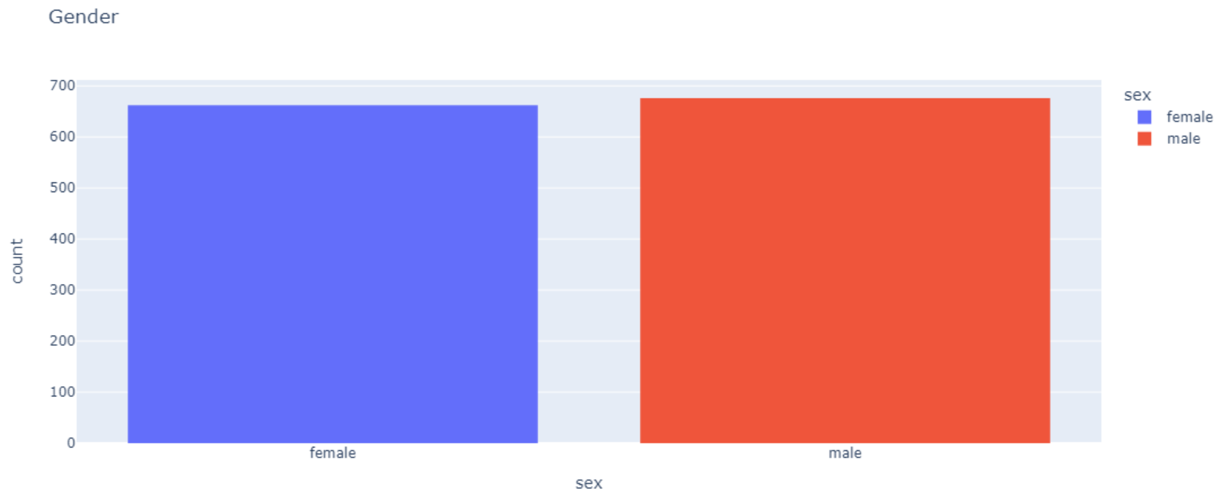
These columns include "smoker", "children", "sex"

```
fig = px.histogram(medical_df, x='smoker', color='sex',
title='smoker')
fig.update_layout(width=700, height=500)
fig.show()
```

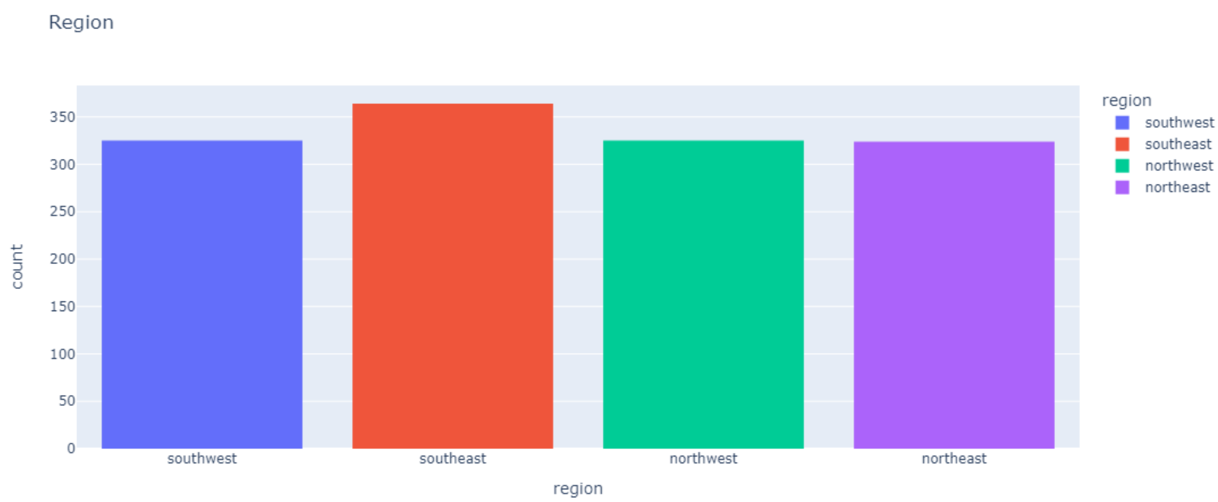


Distribution of Gender

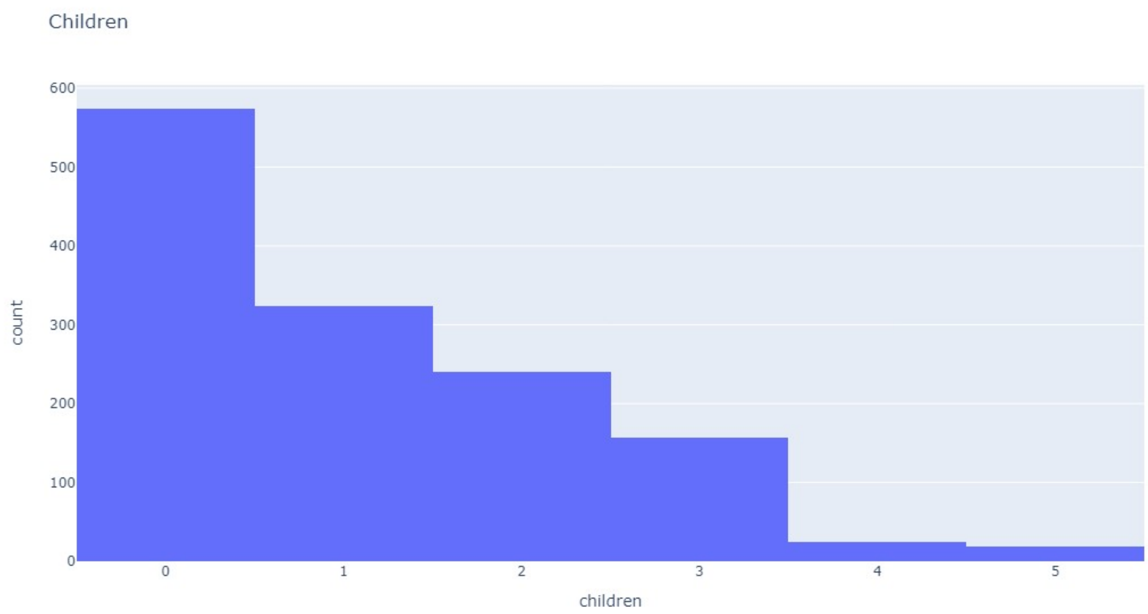
```
fig = px.histogram(medical_df, x='sex', color='sex', title='Gender')
fig.update_layout(width=700, height=500)
fig.show()
```




```
fig = px.histogram(medical_df, x='region', color='region',
title='Region')
fig.update_layout(width=700, height=500)
fig.show()
```




```
fig = px.histogram(medical_df, x='children', title='Children')
fig.update_layout(width=750, height=600)
fig.show()
```



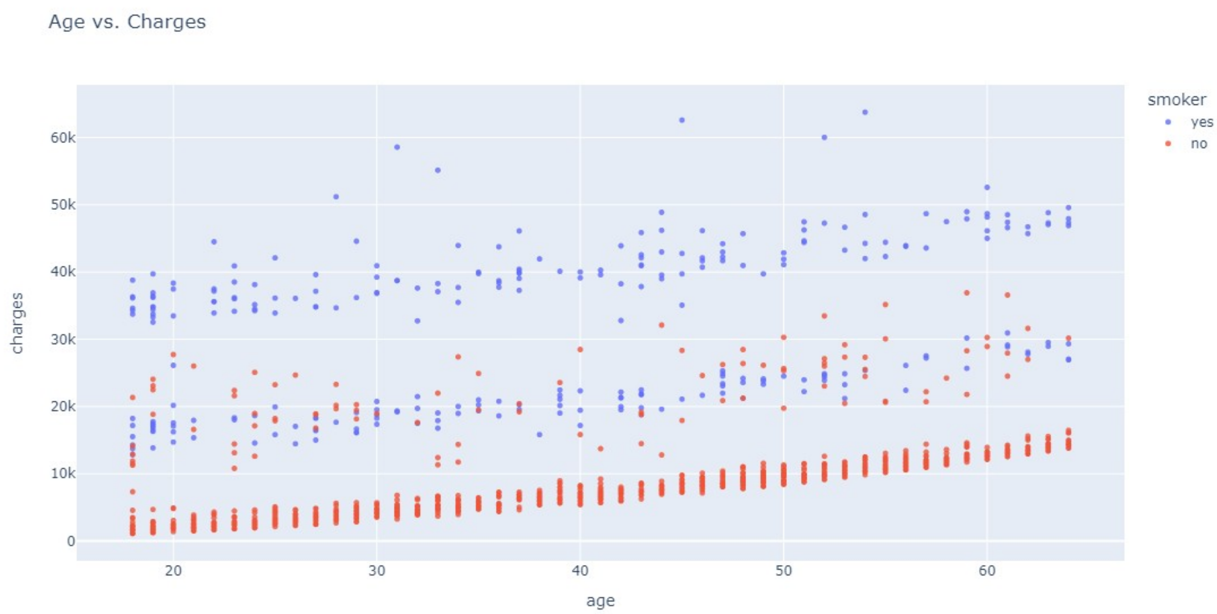
We can visualize the relationship between multiple columns

Age vs Charges

We will use a scatter plot to visualize the difference between "age" and "charges". Every point of the scatter plot represents a customer.

We can also use the smoker column to color the points

```
fig = px.scatter(medical_df,
                 x='age',
                 y='charges',
                 color='smoker',
                 opacity=0.8,
                 hover_data=['sex'],
                 title='Age vs. Charges')
fig.update_traces(marker_size=5)
fig.update_layout(width=750, height=600)
fig.show()
```



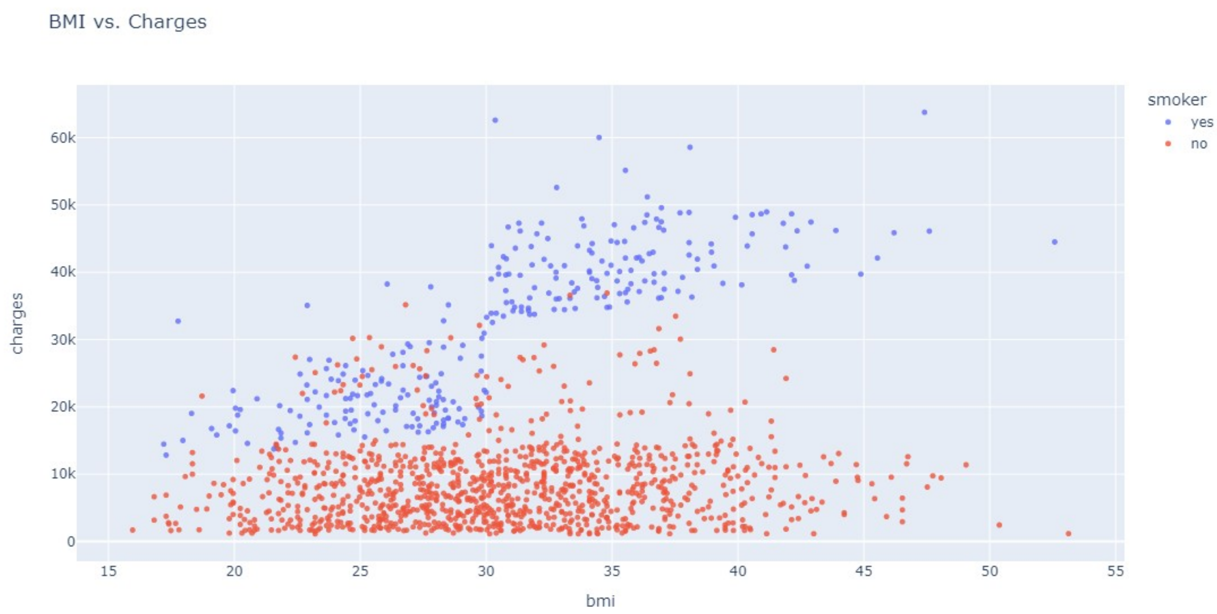
Observations from the chart above;

- The general trend seems to indicate that medical charges increase with age. There is however variations at every age and so we cannot overly rely on age to determine the medical charges
- The data forms three clusters forming a sloping line;
    a. The first lower cluster is presumably the "healthy non-smoker"
    b. The second mid cluster has both smokers and non-smokers. This could be an overlap between "non-smokers with health issues" or "smokers without major medical issues"
    c. The final upper cluster consists of exclusive smokers with issues potentially related to the smoking behaviour

BMI and Charges

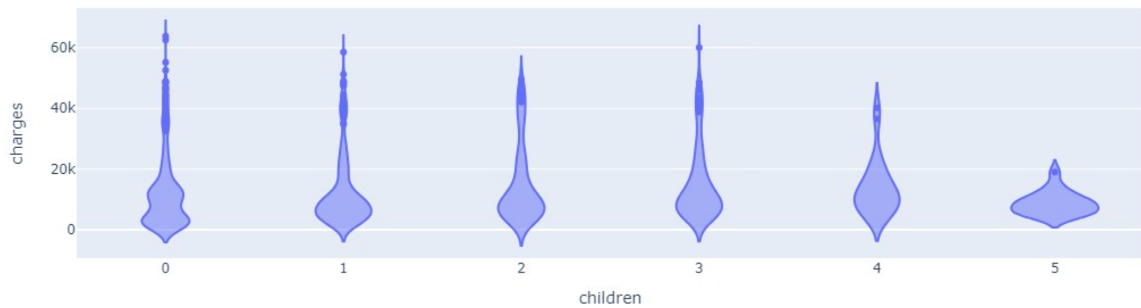
Lets plot a chart to compare BMI to Charges

```
fig = px.scatter(medical_df,
                 x='bmi',
                 y='charges',
                 color='smoker',
                 opacity=0.8,
                 hover_data=['sex'],
                 title='BMI vs. Charges')
fig.update_traces(marker_size=5)
fig.update_layout(width=750, height=600)
fig.show()
```

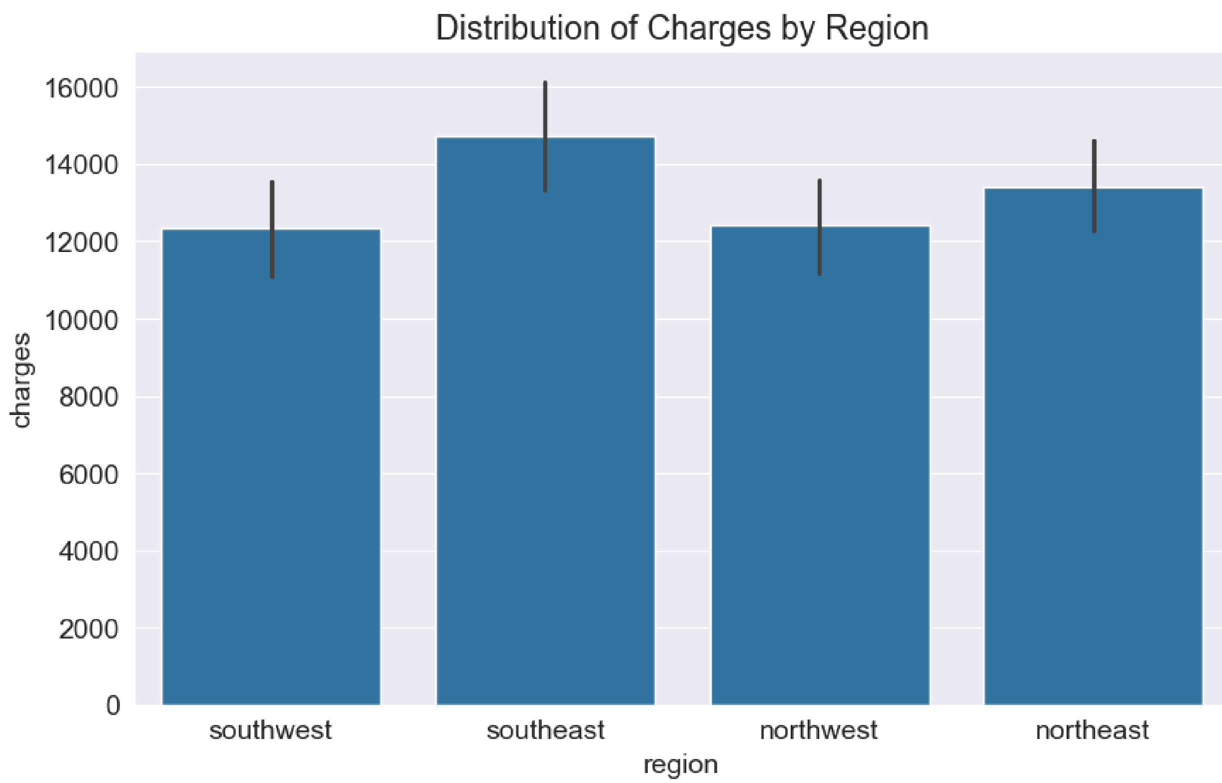


From the plot, it seems that an increase in BMI is not related to an increase in medical charges for non-smokers. But for smokers, there is a significant increase in charges for a BMI greater than 30.

```
px.violin(medical_df, x='children', y='charges')
```




```
ax = sns.barplot(medical_df, x='region', y='charges')
ax.set_title('Distribution of Charges by Region')

Text(0.5, 1.0, 'Distribution of Charges by Region')
```



Correlation

From the EDA, some columns are more closely related to the "charges" than others.

The relationship between different features can be measured using the *correlation coefficient*, which can be calculated using the .corr method

```
medical_df.charges.corr(medical_df.age)
```

```
0.29900819333064765
```

```
medical_df.charges.corr(medical_df.bmi)
```

```
0.19834096883362892
```

```
medical_df.charges.corr(medical_df.children)
```

```
0.06799822684790487
```

To calculate the correlation for categorical columns, they must first be converted into numeric columns

```
# The .map function is used to apply transformations to elements of an iterable
# In our case, we will transform categorical columns to numeric ones using the map function converting "yes" to 1 and "no" to 0
```

```
smoker_values = {'no': 0, 'yes': 1}
smoker_numeric = medical_df.smoker.map(smoker_values)
medical_df.charges.corr(smoker_numeric)
```

```
0.7872514304984772
```

The correlation coefficient is interepreted as follows:

- Correlation ranges from -1 to 1 with 0 indicating no correlation, 1 means a positive correlation and -1 means a negative correlation
- The closer the corr is to the extremes 1 or -1, the stronger the relationship
- The signs indicate the **direction** of the correlation with values between 0 and 1 indicating a positive correlation
- Negative coefficients indicate an inverse relationship between variables

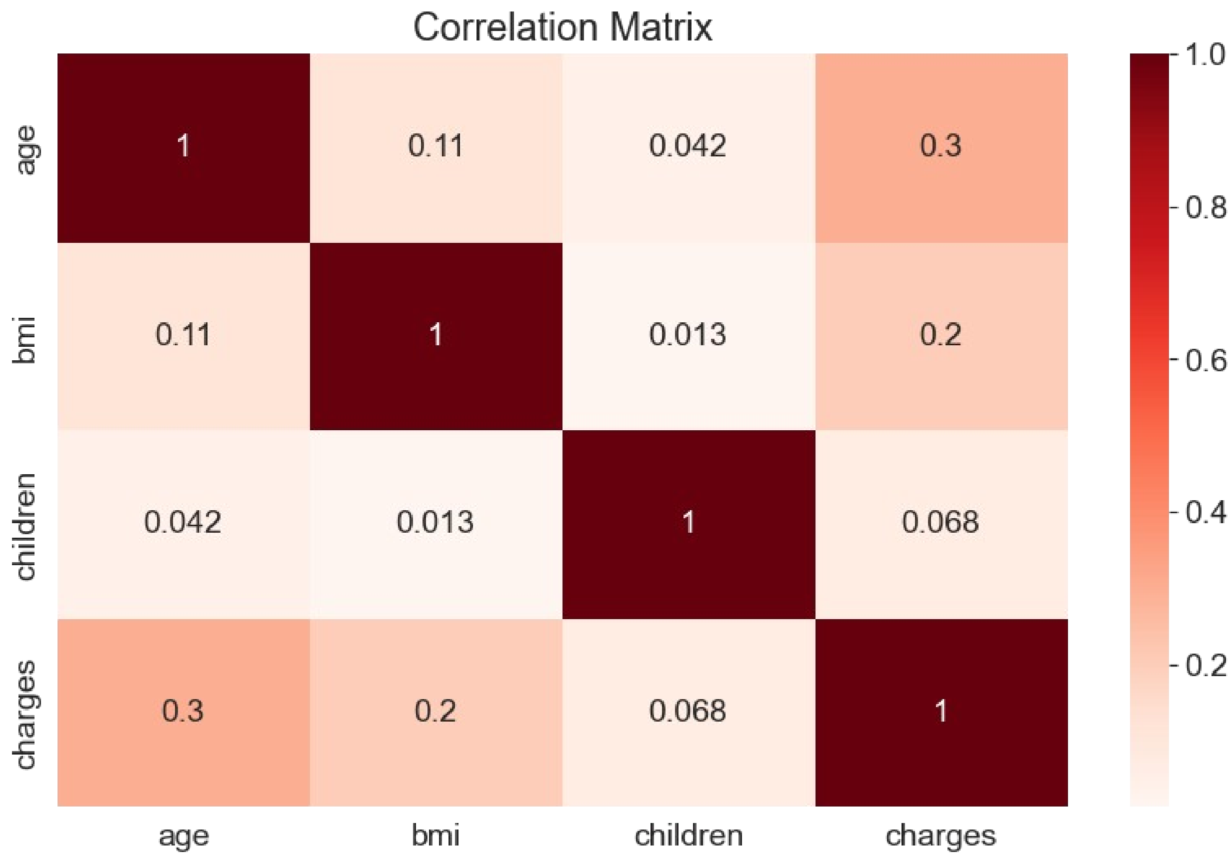
```
numerical_cols = medical_df.select_dtypes(include=[np.number])  # Select numerical columns
correlations = numerical_cols.corr()
correlations
```

	age	bmi	children	charges
age	1.000000	0.109272	0.042469	0.299008
bmi	0.109272	1.000000	0.012759	0.198341
children	0.042469	0.012759	1.000000	0.067998
charges	0.299008	0.198341	0.067998	1.000000

The result of .corr is a correlation matrix that can be visualized as a heatmap

```
sns.heatmap(correlations, cmap='Reds', annot=True)
plt.title('Correlation Matrix')
```

```
Text(0.5, 1.0, 'Correlation Matrix')
```



Note that a high correlation between two variables does not necessarily indicate a cause-effect relationship between features. They could potentially be impacted by another factor. It's therefore to understand the underlying data to determine if indeed there is a correlation.

Linear Regression using a Single Feature

We have noted that "age" and "smoker" have a significant correlation with "charges".

We can try to estimate the value in "charges" using the value of "age" for non-smokers.

Let's first create a Pandas dataframe for the data on non-smokers

```
non_smoker_df = medical_df[medical_df.smoker == 'no']

plt.title('Age vs Charges')
sns.scatterplot(data=non_smoker_df, x='age', y='charges', alpha=0.7,
s=15)
```

```
<Axes: title={'center': 'Age vs Charges'}, xlabel='age',
ylabel='charges'>
```