```
import pandas as pd

# Read the data into a pandas dataframe
data = pd.read_csv("/content/insurance.csv")

# Display the first few rows to check the data
data.head()
```

	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

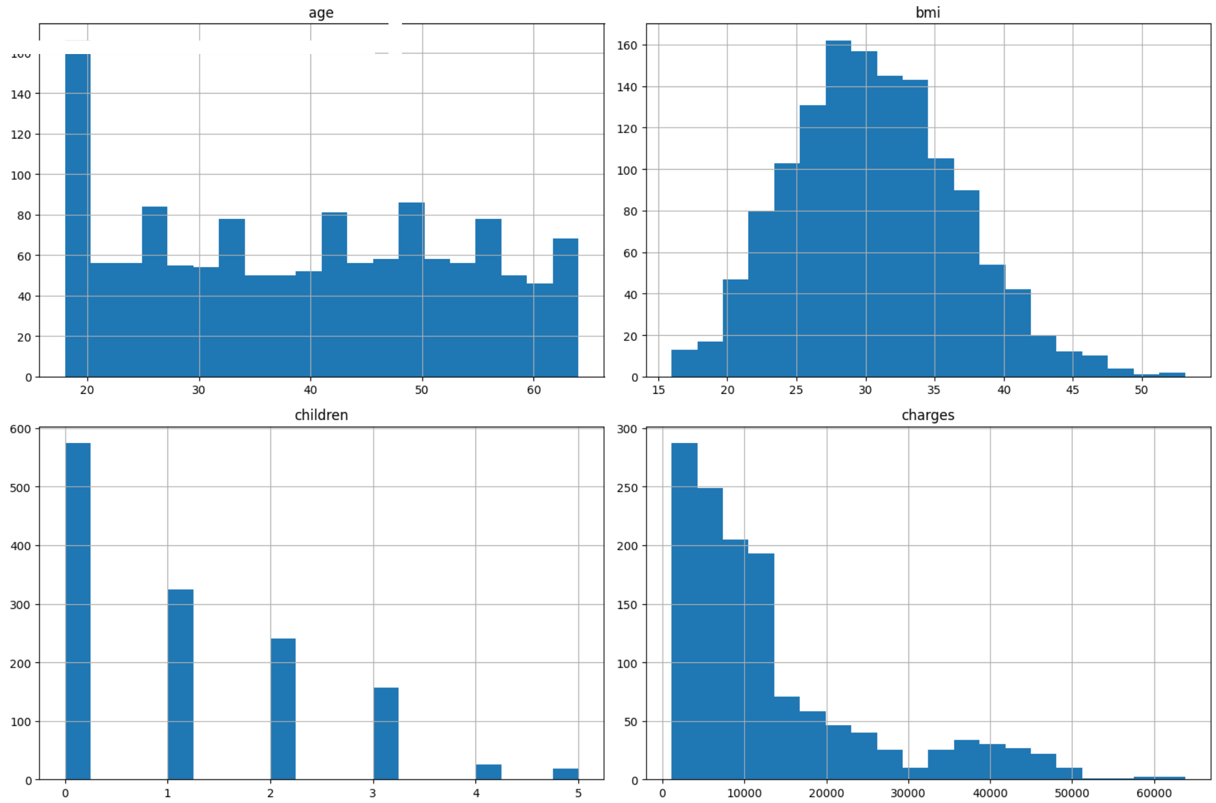
```
# Check the summary of the dataframe
data.info()

# Get descriptive statistics for numerical columns
data.describe()
```

```
<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
```

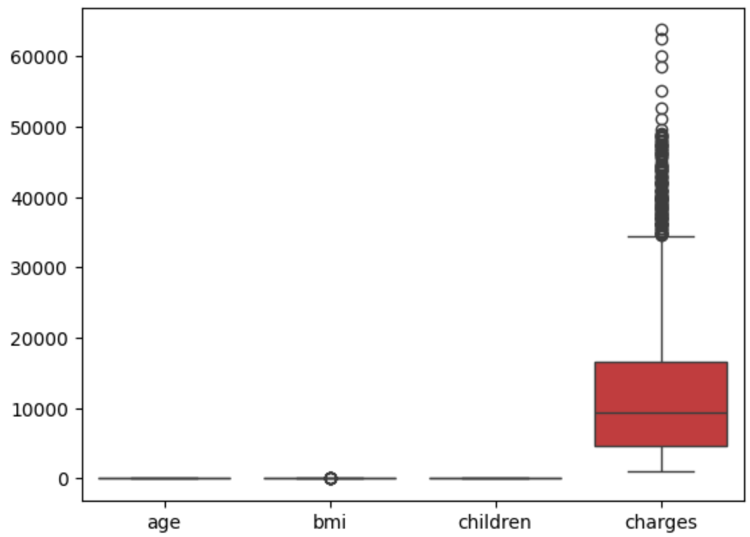
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000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

```
import matplotlib.pyplot as plt
```

```
# Plot histograms for all numerical feature
data.hist(bins=20, figsize=(15, 10))
plt.tight_layout()
plt.show()
```



```
import seaborn as sns

# Boxplot to detect outliers for numerical features
sns.boxplot(data=data[['age', 'bmi', 'children', 'charges']])
plt.show()
```



```
import pandas as pd
from scipy import stats
import numpy as np # Import numpy library

# Z-score method to identify outliers
z_scores = stats.zscore(data[['age', 'bmi', 'children', 'charges']])
abs_z_scores = np.abs(z_scores) # Now np is defined and can be used
outliers = (abs_z_scores > 3).all(axis=1)
outliers_data = data[outliers]

# Display rows with outliers
outliers_data
```

age	sex	bmi	children	smoker	region	charges

```
# Check for missing values
missing_values = data.isnull().sum()

# Display missing values count per column
print(missing_values)
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```