

In [1]:

```
import numpy as np
import pandas as pd
from sklearn import datasets
from scipy import stats
import seaborn as sns
```

In [2]:

```
iris = datasets.load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
df['name'] = df['target'].apply(lambda x : iris.target_names[x])
```

In [3]:

```
df.head()
```

Out[3]:

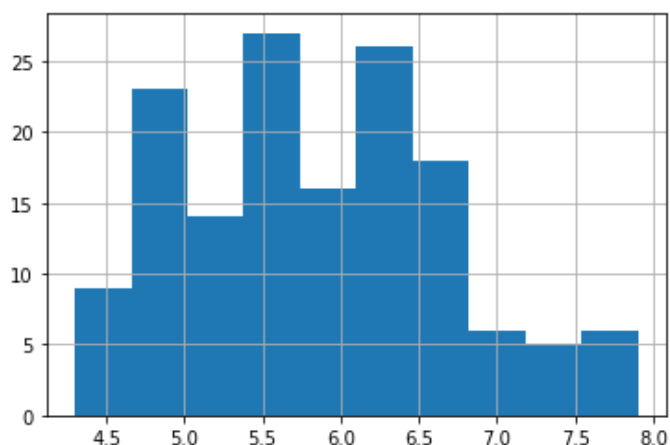
| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target | name |
|---|-------------------|------------------|-------------------|------------------|--------|--------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | 0 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | 0 | setosa |

In [4]:

```
df['sepal length (cm)'].hist()
```

Out[4]:

<AxesSubplot:>



Задача №8.

Для набора данных проведите устранение пропусков для одного (произвольного) числового признака с использованием метода заполнения модой.

Проверим, содержат ли исходные данные пропуски.

In [5]:

```
df.isnull().values.any()
```

Out[5]:

False

Сгенерируем в случайных элементах первой колонки таблицы пропуски.

In [6]:

```
def add_nan(df, col_name):
    df[col_name] = np.where(np.random.randint(2, size=len(df)), np.nan, df[col_name])
    return df
```

In [7]:

```
df1 = df.copy()

df1 = add_nan(df1, 'sepal length (cm)')
df1.head(10)
```

Out[7]:

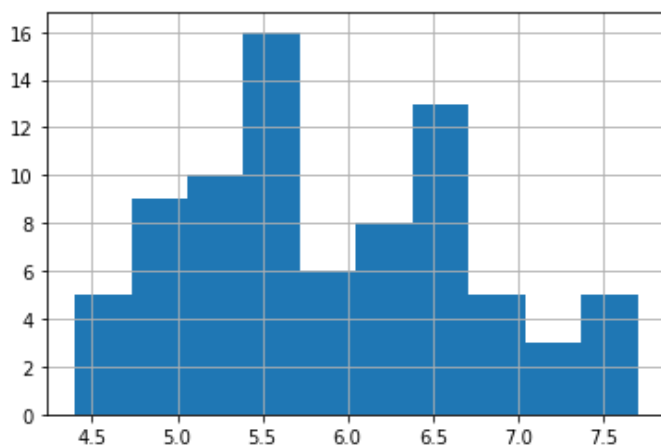
| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target | name |
|---|-------------------|------------------|-------------------|------------------|--------|--------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| 1 | NaN | 3.0 | 1.4 | 0.2 | 0 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | 0 | setosa |
| 5 | NaN | 3.9 | 1.7 | 0.4 | 0 | setosa |
| 6 | 4.6 | 3.4 | 1.4 | 0.3 | 0 | setosa |
| 7 | NaN | 3.4 | 1.5 | 0.2 | 0 | setosa |
| 8 | NaN | 2.9 | 1.4 | 0.2 | 0 | setosa |
| 9 | NaN | 3.1 | 1.5 | 0.1 | 0 | setosa |

In [8]:

```
df1['sepal length (cm)'].hist()
```

Out[8]:

<AxesSubplot:>



In [9]:

```
mode = float(stats.mode(df1['sepal length (cm)']).mode)
print(mode)
```

5.7

In [10]:

```
df1['sepal length (cm)'] = np.where(np.isnan(df1['sepal length (cm)']), mode, df1['sepal length (cm)'])
df1.head(10)
```

Out[10]:

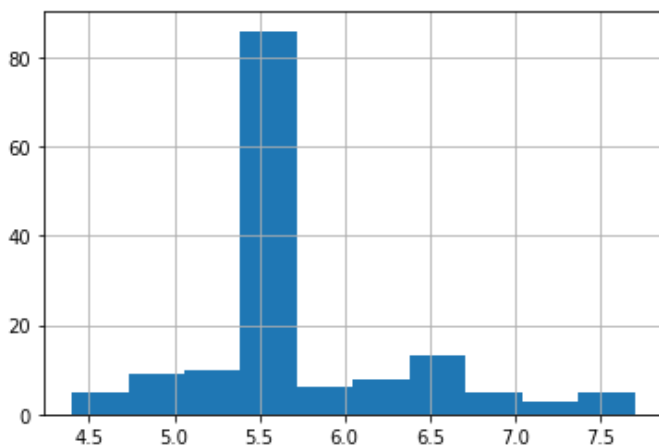
| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target | name |
|---|-------------------|------------------|-------------------|------------------|--------|--------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| 1 | 5.7 | 3.0 | 1.4 | 0.2 | 0 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | 0 | setosa |
| 5 | 5.7 | 3.9 | 1.7 | 0.4 | 0 | setosa |
| 6 | 4.6 | 3.4 | 1.4 | 0.3 | 0 | setosa |
| 7 | 5.7 | 3.4 | 1.5 | 0.2 | 0 | setosa |
| 8 | 5.7 | 2.9 | 1.4 | 0.2 | 0 | setosa |
| 9 | 5.7 | 3.1 | 1.5 | 0.1 | 0 | setosa |

In [11]:

```
df1['sepal length (cm)'].hist()
```

Out[11]:

<AxesSubplot:>



Задача №28.

Для набора данных для одного (произвольного) числового признака проведите обнаружение и замену (найденными верхними и нижними границами) выбросов на основе межквартильного размаха.

In [12]:

```
df2 = df.copy()

quantile_1 = np.percentile(df2['sepal length (cm)'], 25, interpolation='midpoint')
quantile_3 = np.percentile(df2['sepal length (cm)'], 75, interpolation='midpoint')
iqr = quantile_3 - quantile_1 # интерквартильный размах
print(quantile_1, quantile_3)
print(f'{iqr:.{1}f}')
```

5.1 6.4
1.3

In [13]:

```
df2['sepal length (cm)'] = np.where((df2['sepal length (cm)'] < quantile_1), quantile_1,
df2['sepal length (cm)'])
df2['sepal length (cm)'] = np.where((df2['sepal length (cm)'] > quantile_3), quantile_3,
df2['sepal length (cm)'])
df2.head(10)
```

Out[13]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target | name |
|---|-------------------|------------------|-------------------|------------------|--------|--------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| 1 | 5.1 | 3.0 | 1.4 | 0.2 | 0 | setosa |
| 2 | 5.1 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| 3 | 5.1 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| 4 | 5.1 | 3.6 | 1.4 | 0.2 | 0 | setosa |
| 5 | 5.4 | 3.9 | 1.7 | 0.4 | 0 | setosa |
| 6 | 5.1 | 3.4 | 1.4 | 0.3 | 0 | setosa |
| 7 | 5.1 | 3.4 | 1.5 | 0.2 | 0 | setosa |
| 8 | 5.1 | 2.9 | 1.4 | 0.2 | 0 | setosa |
| 9 | 5.1 | 3.1 | 1.5 | 0.1 | 0 | setosa |

In [14]:

```
print(df2['sepal length (cm)'].min(), df2['sepal length (cm)'].max())
```

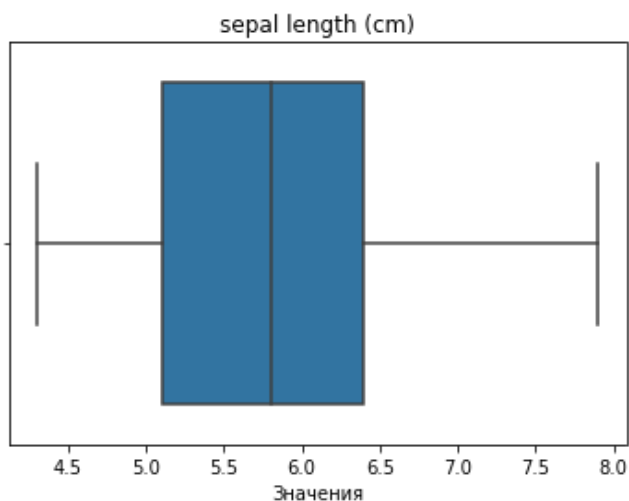
5.1 6.4

Дополнительные требования по группам:

- Для студентов групп ИУ5-23М, ИУ5И-23М - для произвольной колонки данных построить график "Ящик с усами (boxplot)".

In [15]:

```
bplt = sns.boxplot(x=df['sepal length (cm)'])
bplt.axes.set_title('sepal length (cm)')
bplt.axes.set_xlabel('Значения');
```



In [16]:

```
df_min = df['sepal length (cm)'].min()
df_max = df['sepal length (cm)'].max()
df_median = np.median(df['sepal length (cm)'])
```

```
print(f'Минимальное значение: {df_min}')
print(f'Первый квантиль: {quantile_1}')
```

```
print(f'Медиана: {df_median}')
```

```
print(f'Третий квантиль: {quantile_3}')
```

```
print(f'Максимальное значение: {df_max}')
```

Минимальное значение: 4.3

Первый квантиль: 5.1

Медиана: 5.8

Третий квантиль: 6.4

Максимальное значение: 7.9