In [71]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, gaussian_kde
import seaborn as sns
```

In [72]:
```
from sklearn.datasets import load_iris
iris=load_iris()
x1 = np.array(iris.data)
print(iris.feature_names)
sepal_width=x1[:,1]
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [73]:
```
sepal_width
```

Out[73]:
```
array([3.5, 3. , 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1, 3.7, 3.4, 3. ,
       3. , 4. , 4.4, 3.9, 3.5, 3.8, 3.8, 3.4, 3.7, 3.6, 3.3, 3.4, 3. ,
       3.4, 3.5, 3.4, 3.2, 3.1, 3.4, 4.1, 4.2, 3.1, 3.2, 3.5, 3.6, 3. ,
       3.4, 3.5, 2.3, 3.2, 3.5, 3.8, 3. , 3.8, 3.2, 3.7, 3.3, 3.2, 3.2,
       3.1, 2.3, 2.8, 2.8, 3.3, 2.4, 2.9, 2.7, 2. , 3. , 2.2, 2.9, 2.9,
       3.1, 3. , 2.7, 2.2, 2.5, 3.2, 2.8, 2.5, 2.8, 2.9, 3. , 2.8, 3. ,
       2.9, 2.6, 2.4, 2.4, 2.7, 2.7, 3. , 3.4, 3.1, 2.3, 3. , 2.5, 2.6,
       3. , 2.6, 2.3, 2.7, 3. , 2.9, 2.9, 2.5, 2.8, 3.3, 2.7, 3. , 2.9,
       3. , 3. , 2.5, 2.9, 2.5, 3.6, 3.2, 2.7, 3. , 2.5, 2.8, 3.2, 3. ,
       3.8, 2.6, 2.2, 3.2, 2.8, 2.8, 2.7, 3.3, 3.2, 2.8, 3. , 2.8, 3. ,
       2.8, 3.8, 2.8, 2.8, 2.6, 3. , 3.4, 3.1, 3. , 3.1, 3.1, 3.1, 2.7,
       3.2, 3.3, 3. , 2.5, 3. , 3.4, 3. ])
```

In [74]:
```
def generate_normal_qq_table(arr):
    n = len(arr)
    q_table = pd.DataFrame()
    q_table['Ordered Observations'] = sorted(arr)
    q_table['Probability Levels'] = [ (j-0.5)/n for j in range(1,n+1)]
    q_table['Theoretical Quantile'] = norm.ppf(q_table['Probability Levels'])
    return q_table
def qq_plot(arr,ax,**kwargs):
    q_table = generate_normal_qq_table(arr)
    ax.scatter(q_table['Theoretical Quantile'], q_table['Ordered Observations'])
    ax.set_xlabel('Theoretical Quantile')
    ax.set_ylabel('Ordered Observations')
    ax.set_title('QQ-Plot')
def plot_hist(arr,ax):
    df = pd.DataFrame(arr)
    df.plot.kde(ax = ax, legend = False)
    df.plot.hist(ax = ax, density = True, legend = False, rwidth=0.9)
```

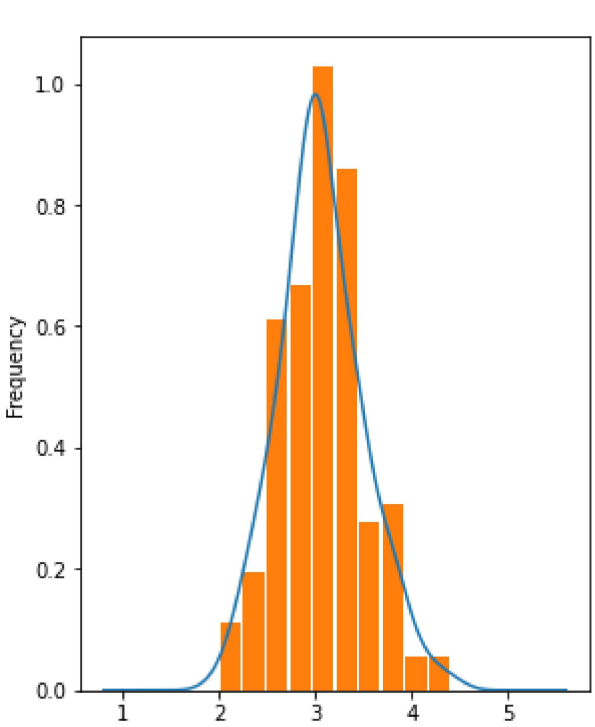
In [75]:
```
q_table = generate_normal_qq_table(sepal_width)
q_table.head()
```

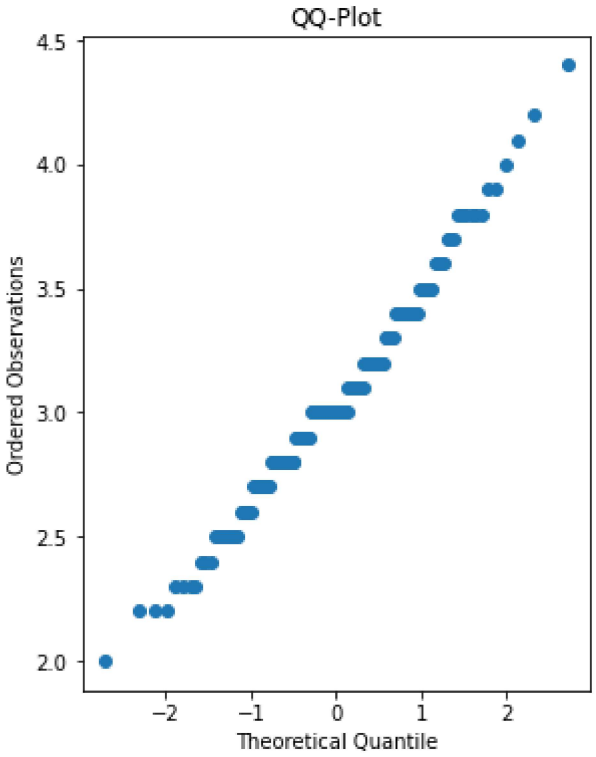
Out[75]:

	Ordered Observations	Probability Levels	Theoretical Quantile
0	2.0	0.003333	-2.713052
1	2.2	0.010000	-2.326348

	Ordered Observations	Probability Levels	Theoretical Quantile
2	2.2	0.016667	-2.128045
3	2.2	0.023333	-1.989313
4	2.3	0.030000	-1.880794

In [76]:
```
fig, axs = plt.subplots(figsize = (10,6),nrows=1,ncols=2)
plot_hist(sepal_width,axs[0])
qq_plot(sepal_width,axs[1])
```





Hypothesis:

H0: correlation >= tolerance

HA: correlation < tolerance

In [77]:
```
correlation= np.corrcoef(q_table['Theoretical Quantile'],q_table['Ordered Observations'
correlation
```

Out[77]:
```
array([[1.        , 0.99251131],
       [0.99251131, 1.        ]])
```

In [78]:
```
tolerance = 0.9901 # for sample size 150 .
if correlation[0,1] > tolerance:
    print('Sample looks Normal (fail to reject H0) according to Correlation Coeeficient
else:
    print('Sample does not look Normal (reject H0) according to Correlation Coeeficient
```

```
Sample looks Normal (fail to reject H0) according to Correlation Coeeficient Test
```

In [79]:

```
from scipy.stats import norm, shapiro, kstest
alpha = 0.05
stat, p = shapiro(sepal_width)
print("Shapiro-Wilk Test:")
print(f'--- Statistics={stat:.3f}, p={p:.3f}')
if p > alpha:
    print('--- Sample looks Normal (fail to reject H0) according to Shapiro')
else:
    print('--- Sample does not look Normal (reject H0) according to Shapiro')
```

```
Shapiro-Wilk Test:
--- Statistics=0.985, p=0.101
--- Sample looks Normal (fail to reject H0) according to Shapiro
```

We accept null hypothesis as per correlation test where the tolerance level is lower than the correlation coefficient and as the p value is greater than alpha (0.05) obtained from shapiro test.

In [ ]: