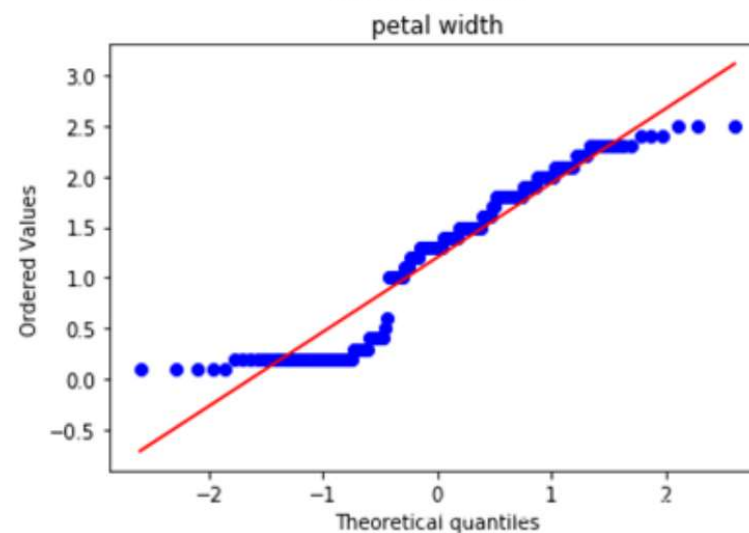
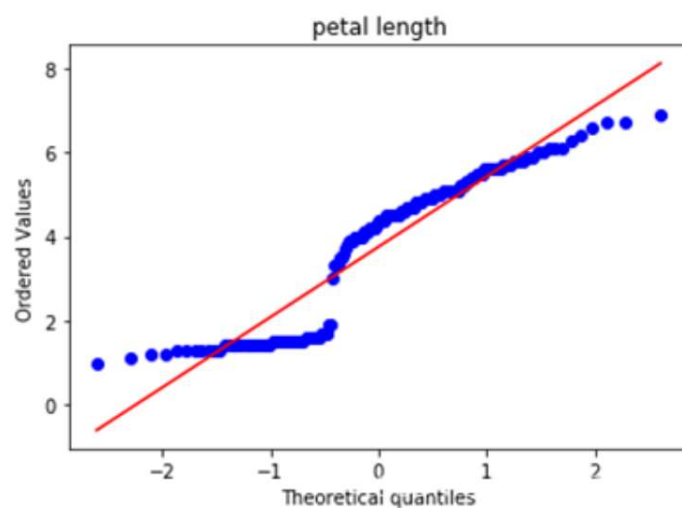
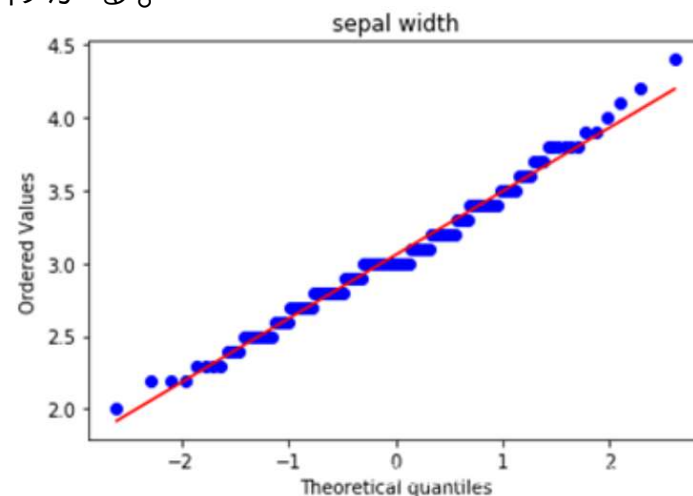
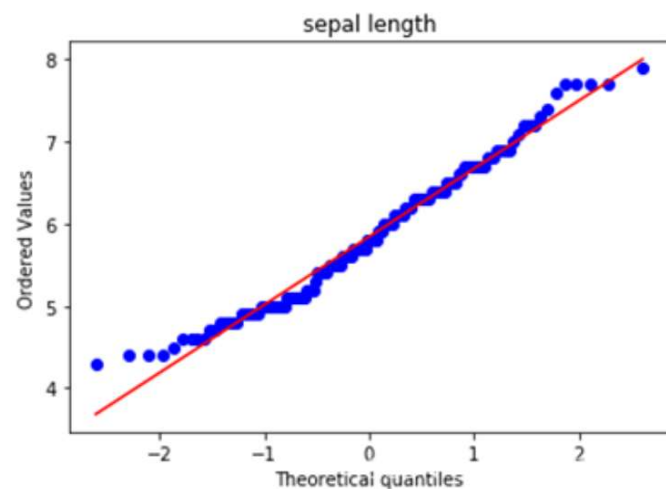


**このデータセットは、実験要件に従い、80%をトレーニング用、残りの20%をテスト用とするランダムな分割を行った。**

実験データの4つの特徴変数はすべて連続型の特徴変数であるので、ガウス分布ブレインベイズを使う必要があれば、彼がガウス分布に合致しているかどうかだけを検証すればよい。ここで、データがガウス分布に合致しているかどうかをQ-Qプロットで確認する。もし、データがガウス分布に合致していれば、Q-Qプロットは $y=x$ 線の上に落ちる傾向があり、次のコードでプロットされる：

```
1 | from scipy.stats import probplot
2 | probplot(data,dist='norm',plot=plt)
3 | plt.title('sepal length')
4 | plt.show()
```

プロットした結果を、4つの特徴ベクトルのQ-Qプロットについて、順に以下に示す。4つの特徴ベクトルの全体的な傾向は、ほぼガウス分布に沿っており、ガウス分布の平明なベイズ分類法を利用することができることがわかる。



課題1：sklearnライブラリのGaussianNB()メソッドを用いた分類器の学習、コードは以下の通り：

```
# ガウスナイーブベイズによる計算
clf = GaussianNB()
clf.fit(X_train, y_train)
```

課題2：テストセットから最後のデータを選んで予測し、判別精度、分類器によって計算された分類結果、確率を観察する

判別精度：0.967

[2]

期待される確率の値：[[1.63380783e-232 2.18878438e-006 9.99997811e-001]]

上記の例の予測結果から、判別精度は0.967

# 更なる分析結果と考察

評価指標は、分類器のaccuracy、recall、F1指標値を評価指標として選択し、sklearnライブラリのaccuracy\_score()、precision\_score()、recall\_score()、f1\_score()メソッドを用い、以下のコードで計算した。

```
# アセスメント
y_pred = clf.predict(X_test)
print(precision_score(y_test, y_pred, average=None))
print(recall_score(y_test, y_pred, average=None))
print(f1_score(y_test, y_pred, average=None))
```

この計算により、2値データセットでは以下のような結果が得られる:

ガウス分布ナীবベース分類器			
	類別0	類別 1	類別 2
精度 P	1	0.92857	1
回収率R	1	1	0.83333
F1-Score	1	0.96296	0.90909

カテゴリ0が最も大きな事後確率値に相当することがわかるので、カテゴリ0が最適な結果であると考えることができる。

## 感想:

本実験では、Irisトリプル分類データセットをナイーブベイズで分類し、生成したモデルの性能を評価するために、Precision、Recall、F1メトリック値を算出することを主眼とした。得られた分類モデルは、より優れた性能を持ち、より正確にIrisトリプルの分類タスクを実行することができる。

この例から、離散データに対するプレーンベイズでは、その分布をガウス分布であるかのように扱うことがわかる。その後、計算が行われる。