Ideal value of pH for various crops

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
cropph=pd.read_csv('C:\\Users\\Admin\\Downloads\\cropph.csv')
cropph.head()
```

Out[1]:

	Crop	Ph
0	Bajra	3.0
1	Banana	6.5
2	Barley	3.0
3	Bean	5.5
4	Black pepper	6.4

In [2]:

```
cropph.columns
```

Out[2]:

```
Index(['Crop', ' Ph'], dtype='object')
```

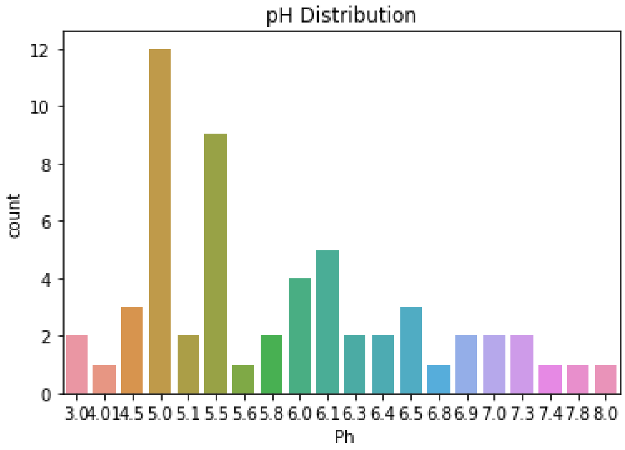
In [3]:

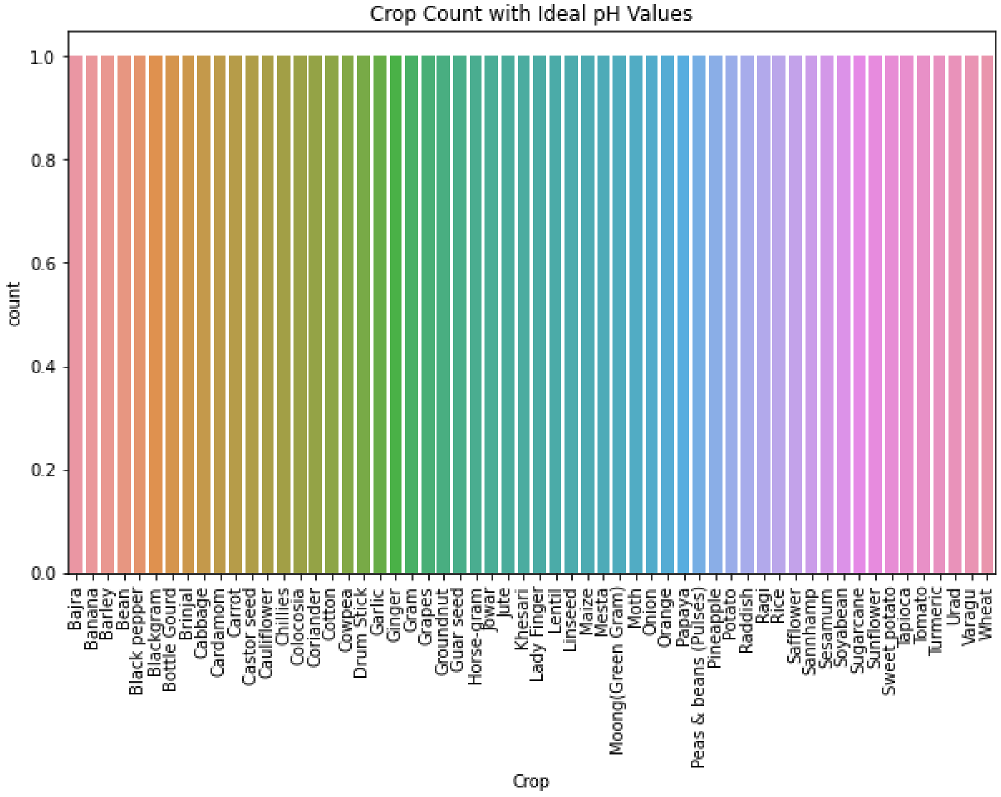
```
cropph.describe()
```

Out[3]:

	Ph
count	58.000000
mean	5.727759
std	1.014469
min	3.000000
25%	5.000000
50%	5.550000
75%	6.375000
max	8.000000

In [4]:
```
# Visualize the distribution of pH values
sns.countplot(data=croppH, x=' Ph')
plt.title('pH Distribution')
plt.show()

# Visualize the count of crops with ideal pH values
plt.figure(figsize=(10, 6))
sns.countplot(data=croppH, x='Crop')
plt.title('Crop Count with Ideal pH Values')
plt.xticks(rotation=90)
plt.show()
```





Actual Crop Yield based on Various Factors

In [5]:

```
cropvaries=pd.read_csv("C:\\Users\\Admin\\Downloads\\cropproductiononvariousfactors.csv")
cropvaries.head()
```

Out[5]:

	Rainfall	Temperature	Ph	Crop	Production	Cropconversion
0	400.15082	20.0	3.0	Bajra	0.00690	Bajra
1	400.16331	20.0	3.2	Bajra	0.00747	Bajra
2	400.16387	20.0	3.2	Bajra	0.00749	Bajra
3	400.17967	20.0	3.2	Bajra	0.00822	Bajra
4	400.19582	20.0	3.2	Bajra	0.00895	Bajra

In [6]:

```
cropvaries.columns
```

Out[6]:

```
Index(['Rainfall', 'Temperature', 'Ph', 'Crop', 'Production',
       'Cropconversion'],
      dtype='object')
```

In [7]:

```
# Get the summary statistics of the dataset
print(cropvaries.describe())
```

```
            Rainfall    Temperature             Ph     Production
count  189232.000000  189232.000000  189232.000000  189232.000000
mean      693.417573      25.262492       6.337846       5.262677
std       288.988419       4.585660       0.788959      14.441616
min       100.003400       7.000000       3.000000       0.000340
25%       516.000000      22.026830       5.800000       0.584420
50%       628.702885      25.000000       6.300000       1.166670
75%       770.000000      28.215000       6.900000       2.989630
max      3000.000000      39.045000       8.800000     955.750340
```

In [8]:

```
# Check the data types of each column
print(cropvaries.dtypes)
```

```
Rainfall          float64
Temperature       float64
Ph                float64
Crop               object
Production        float64
Cropconversion     object
dtype: object
```

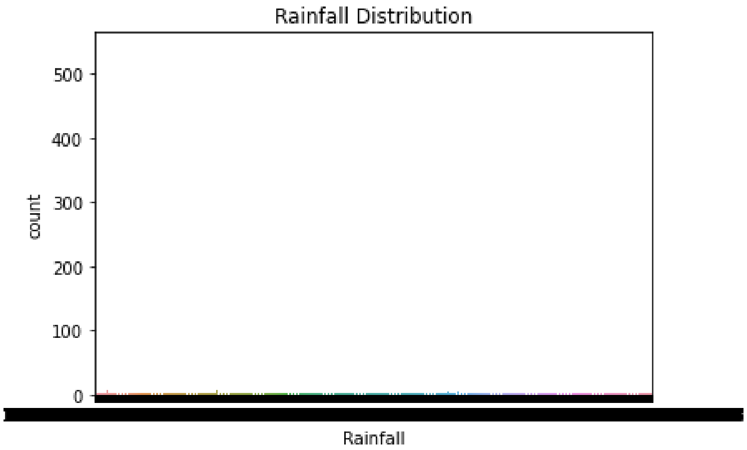
In [9]:

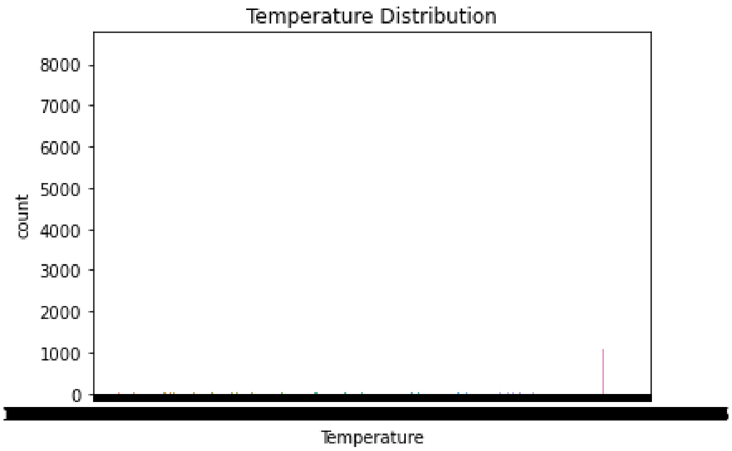
```
# Check for missing values
print(cropvaries.isnull().sum())
```

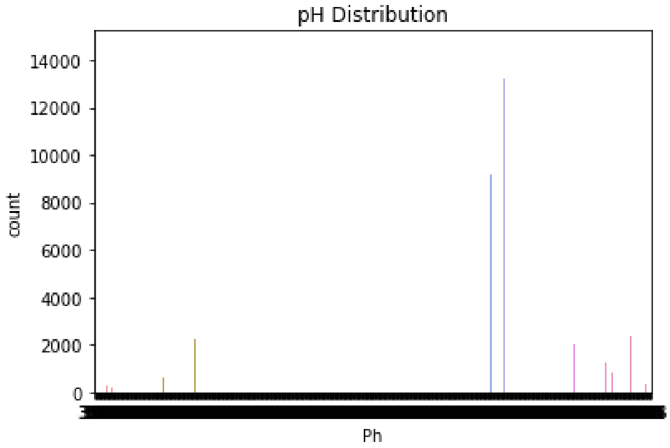
```
Rainfall          0
Temperature       0
Ph                0
Crop              0
Production        0
Cropconversion    0
dtype: int64
```

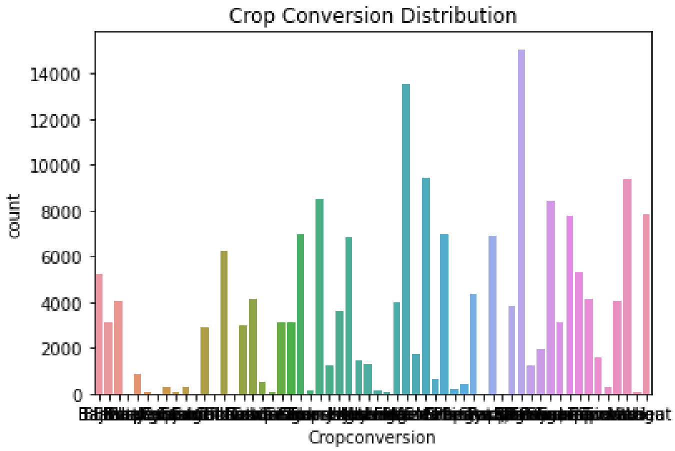
In [10]:

```
# Visualize the distribution of 'Rainfall', 'Temperature', 'Ph', and 'Cropconversion'
sns.countplot(data=cropvaries, x='Rainfall')
plt.title('Rainfall Distribution')
plt.show()

sns.countplot(data=cropvaries, x='Temperature')
plt.title('Temperature Distribution')
plt.show()

sns.countplot(data=cropvaries, x='Ph')
plt.title('pH Distribution')
plt.show()

sns.countplot(data=cropvaries, x='Cropconversion')
plt.title('Crop Conversion Distribution')
plt.show()
```


Rainfall Distribution
count
500
400
300
200
100
0
Rainfall

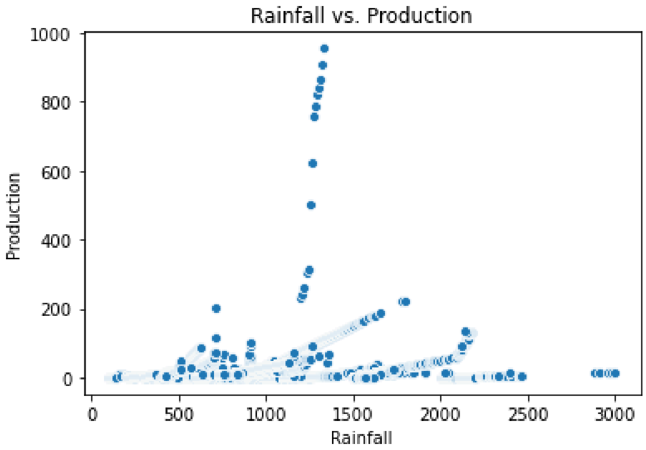

Temperature Distribution
count
8000
7000
6000
5000
4000
3000
2000
1000
0
Temperature


pH Distribution
count
14000
12000
10000
8000
6000
4000
2000
0
Ph


Crop Conversion Distribution
count
14000
12000
10000
8000
6000
4000
2000
0
Cropconversion

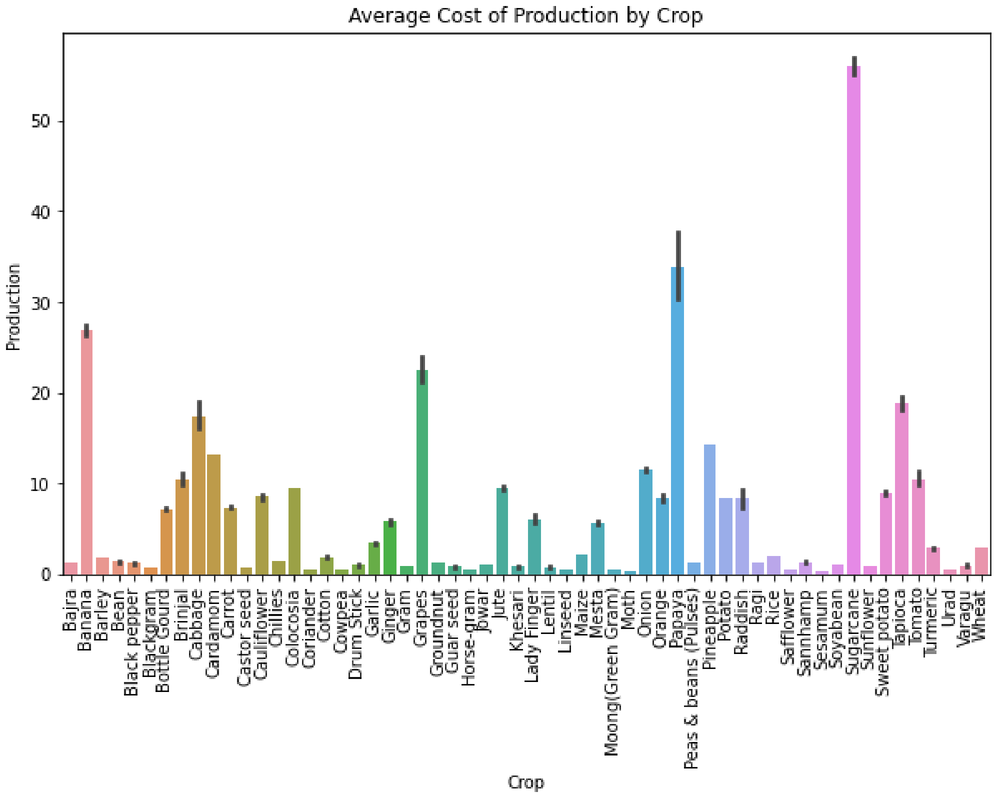
In [11]:

```
# Visualize the relationship between 'Rainfall' and 'Production' using a scatter plot
sns.scatterplot(data=cropvaries, x='Rainfall', y='Production')
plt.title('Rainfall vs. Production')
plt.show()
```



In [12]:

```
# Visualize the average cost of production by crop
plt.figure(figsize=(10, 6))
sns.barplot(data=cropvaries, x='Crop', y='Production')
plt.title('Average Cost of Production by Crop')
plt.xticks(rotation=90)
plt.show()
```



State Wise Cost of Production

In [13]:

```
crop_yeild=pd.read_csv("C:\\Users\\Admin\\Downloads\\state_wise_crop_production.csv")
crop_yeild.head()
```

Out[13]:

	Crop	State	Cost of Cultivation (`/Hectare) A2+FL	Cost of Cultivation (`/Hectare) C2	Cost of Production (`/Quintal) C2	Yield (Quintal/ Hectare)
0	ARHAR	Uttar Pradesh	9794.05	23076.74	1941.55	9.83
1	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
2	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59
3	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
4	ARHAR	Maharashtra	17130.55	25270.26	2775.80	8.72

In [14]:

```
crop_yeild.columns
```

Out[14]:

```
Index(['Crop', 'State', 'Cost of Cultivation (`/Hectare) A2+FL',
       'Cost of Cultivation (`/Hectare) C2',
       'Cost of Production (`/Quintal) C2', 'Yield (Quintal/ Hectare) '],
      dtype='object')
```

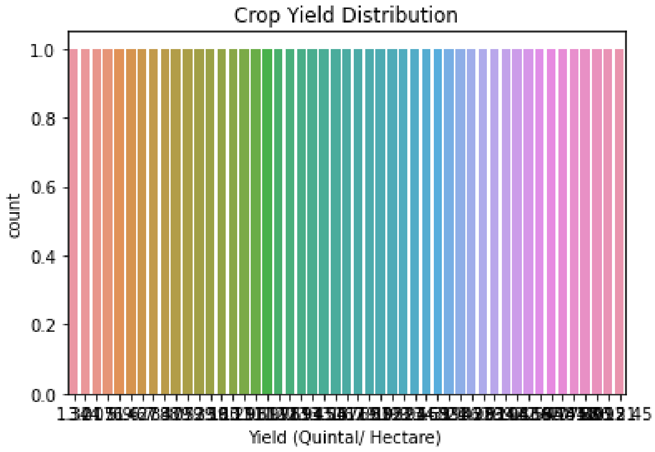
In [15]:

```
# Get the summary statistics of the dataset
print(crop_yeild.describe())

# Check the data types of each column
print(crop_yeild.dtypes)

# Check for missing values
print(crop_yeild.isnull().sum())
```

```
       Cost of Cultivation (`/Hectare) A2+FL  \
count                              49.000000   
mean                            20363.537347   
std                             13561.435306   
min                              5483.540000   
25%                             12774.410000   
50%                             17022.000000   
75%                             24731.060000   
max                             66335.060000   

       Cost of Cultivation (`/Hectare) C2  Cost of Production (`/Quintal) C2  \
count                           49.000000                          49.000000   
mean                         31364.666735                        1620.537755   
std                          20095.783569                        1104.990472   
min                           7868.640000                          85.790000   
25%                          19259.840000                         732.620000   
50%                          25909.050000                        1595.560000   
75%                          35423.480000                        2228.970000   
max                          91442.630000                        5777.480000   

       Yield (Quintal/ Hectare)  
count                 49.000000  
mean                  98.086735  
std                  245.293123  
min                    1.320000  
25%                    9.590000  
50%                   13.700000  
75%                   36.610000  
max                 1015.450000  
Crop                                      object
State                                     object
Cost of Cultivation (`/Hectare) A2+FL    float64
Cost of Cultivation (`/Hectare) C2       float64
Cost of Production (`/Quintal) C2        float64
Yield (Quintal/ Hectare)                 float64
dtype: object
Crop                                     0
State                                    0
Cost of Cultivation (`/Hectare) A2+FL    0
Cost of Cultivation (`/Hectare) C2       0
Cost of Production (`/Quintal) C2        0
Yield (Quintal/ Hectare)                 0
dtype: int64
```

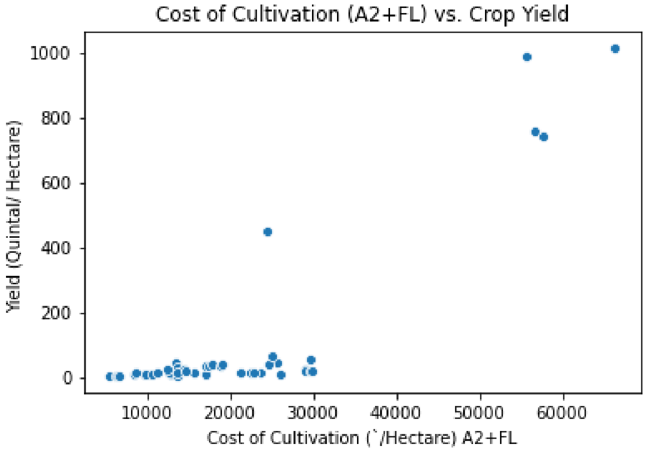
In [16]:

```
# Visualize the distribution of'Yield (Quintal/ Hectare)'

sns.countplot(data=crop_yeild, x='Yield (Quintal/ Hectare) ')
plt.title('Crop Yield Distribution')
plt.show()
```



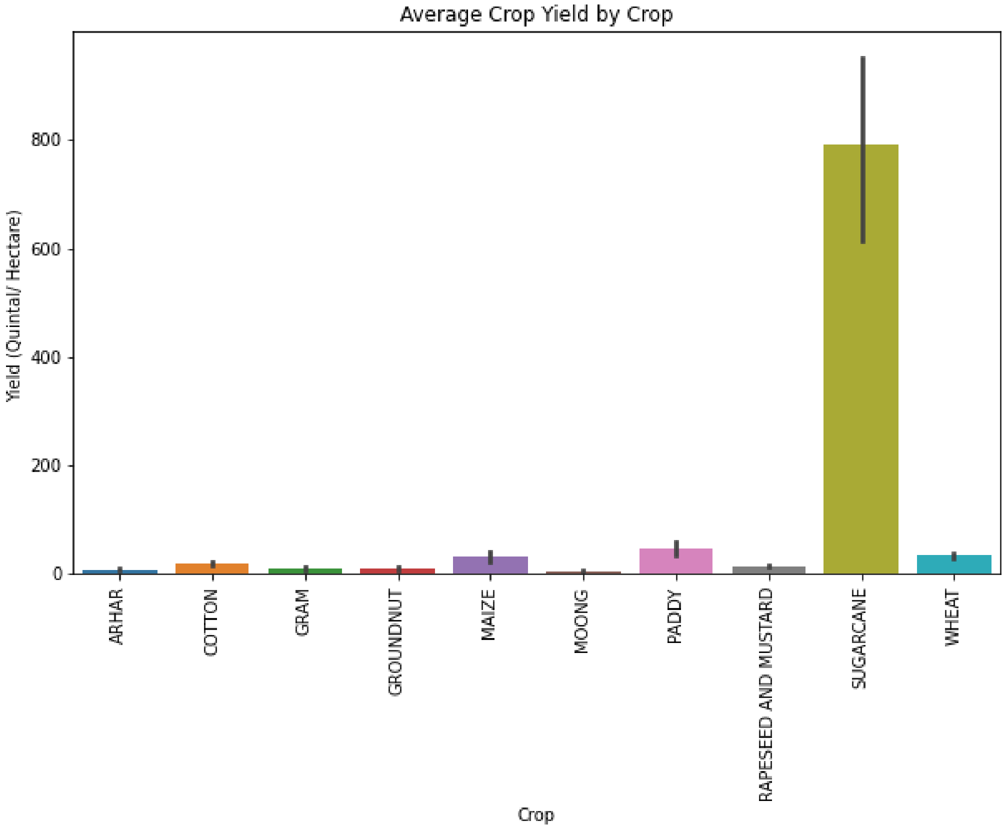
In [17]:

```
# Visualize the relationship between 'Cost of Cultivation (`/Hectare) A2+FL' and 'Yield (Quintal/ Hectare)'
sns.scatterplot(data=crop_yeild, x='Cost of Cultivation (`/Hectare) A2+FL', y='Yield (Quintal/ Hectare) ')
plt.title('Cost of Cultivation (A2+FL) vs. Crop Yield')
plt.show()
```



In [18]:

```python
# Visualize the average yield by crop
plt.figure(figsize=(10, 6))
sns.barplot(data=crop_yeild, x='Crop', y='Yield (Quintal/ Hectare) ')
plt.title('Average Crop Yield by Crop')
plt.xticks(rotation=90)
plt.show()
```


Average Crop Yield by Crop
Yield (Quintal/ Hectare)
ARHAR
COTTON
GRAM
GROUNDNUT
MAIZE
MOONG
PADDY
RAPESEED AND MUSTARD
SUGARCANE
WHEAT
Crop