Module 2: Exploratory data analysis and data visualization

In [1]:

```
import numpy as n
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
df = p.read_csv("crop.csv") #Load dataset
```

In [4]:

```
df = df.dropna() #Drop NULL values
```

In [5]:

```
df.columns #Return the columns
```

Out[5]:

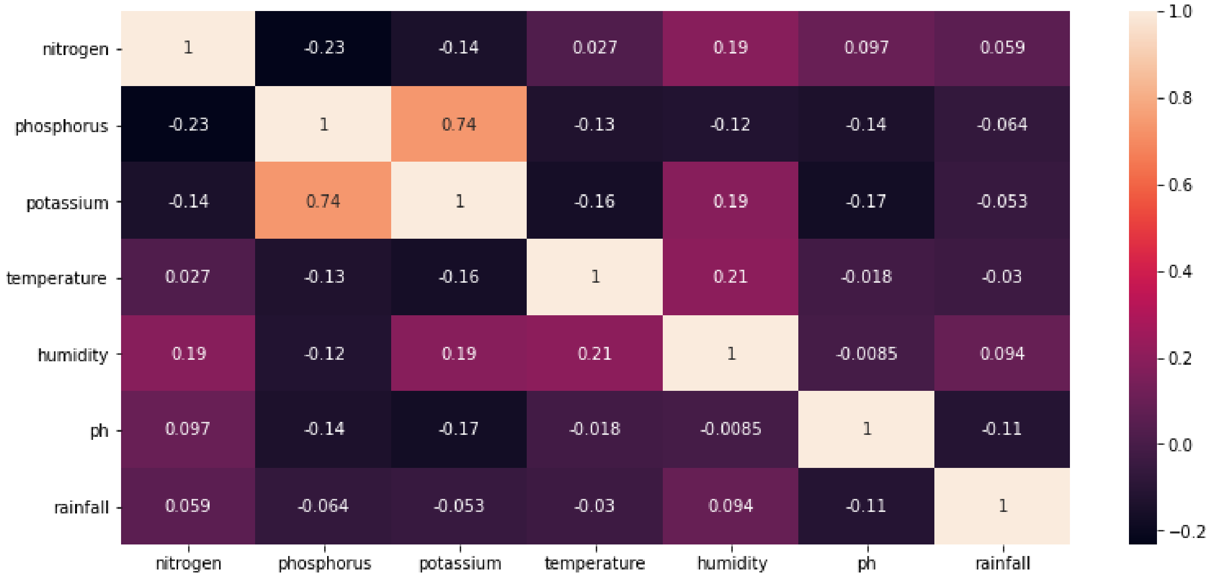
```
Index(['nitrogen', 'phosphorus', 'potassium', 'temperature', 'humidity', 'ph',
       'rainfall', 'label'],
      dtype='object')
```

In [6]:

```
fig, ax = plt.subplots(figsize=(13,6))
s.heatmap(df.corr(), ax=ax, annot=True) #Heatmap to show correlation between feature
```

Out[6]:

```
<AxesSubplot:>
```

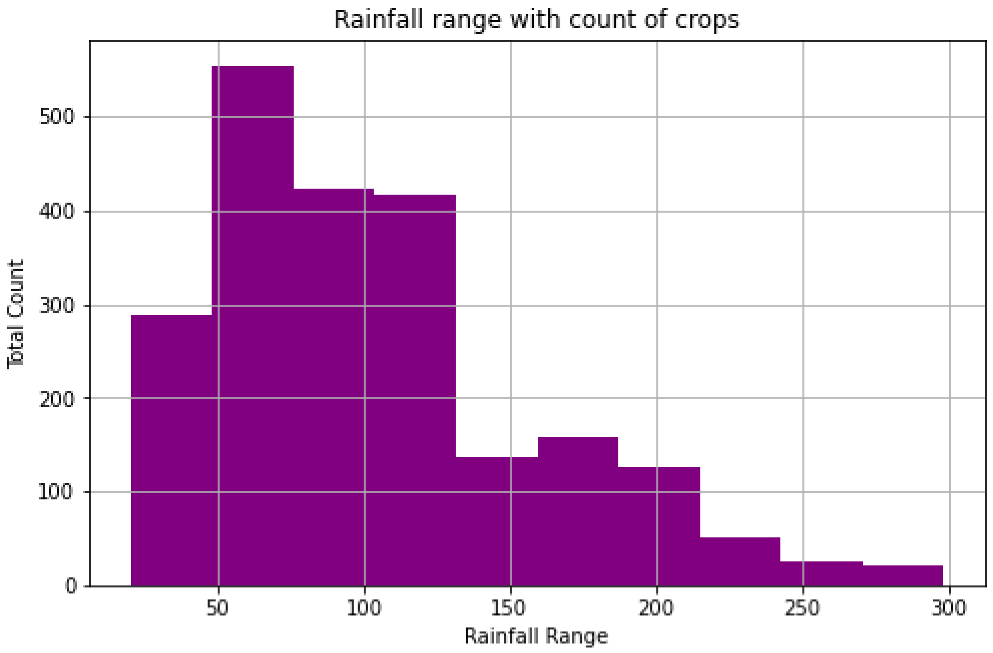


In [7]:

```
df['rainfall'].hist(figsize=(8,5), color='purple')
plt.title('Rainfall range with count of crops')
plt.xlabel('Rainfall Range')
plt.ylabel('Total Count')
```

Out[7]:

```
Text(0, 0.5, 'Total Count')
```

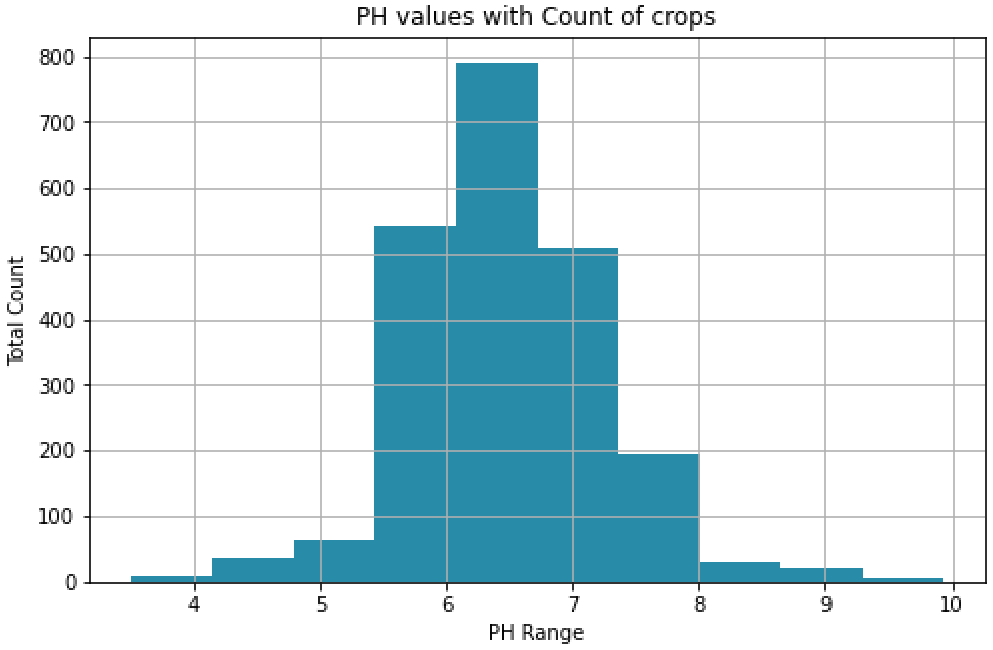


In [8]:

```
df['ph'].hist(figsize=(8,5), color='#288BA8')
plt.title('PH values with Count of crops')
plt.xlabel('PH Range')
plt.ylabel('Total Count')
```

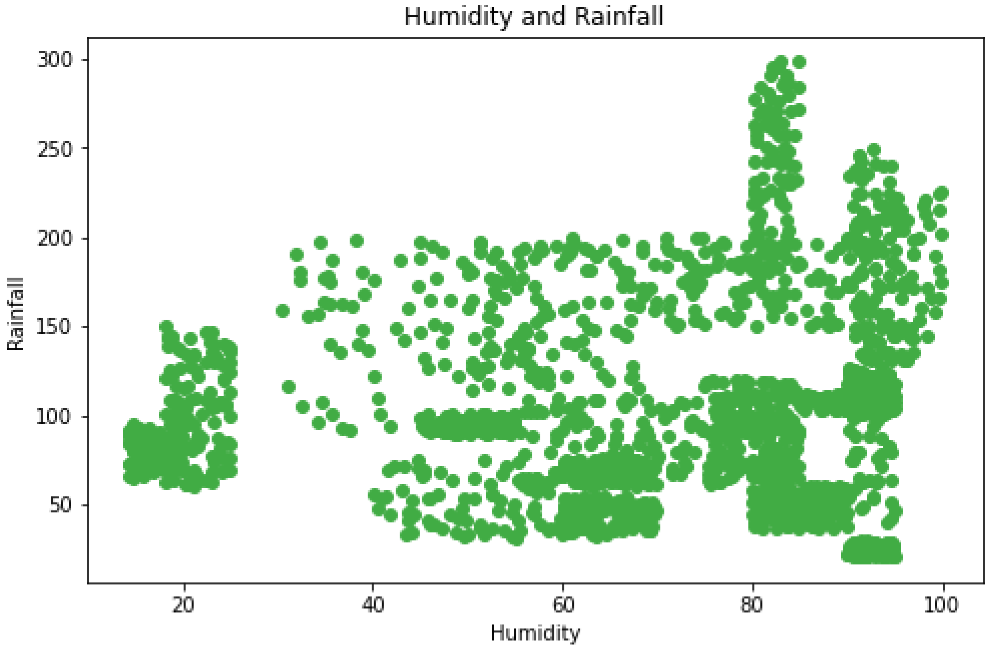
Out[8]:

```
Text(0, 0.5, 'Total Count')
```



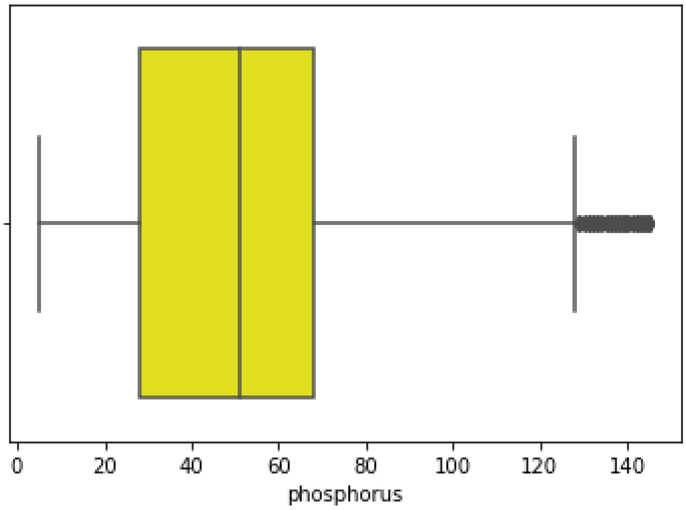
In [9]:

```
fig, ax = plt.subplots(figsize=(8,5))

ax.scatter(df['humidity'],df['rainfall'],color='#41AC44')

ax.set_title('Humidity and Rainfall')
ax.set_xlabel('Humidity')
ax.set_ylabel('Rainfall')

plt.show()
```



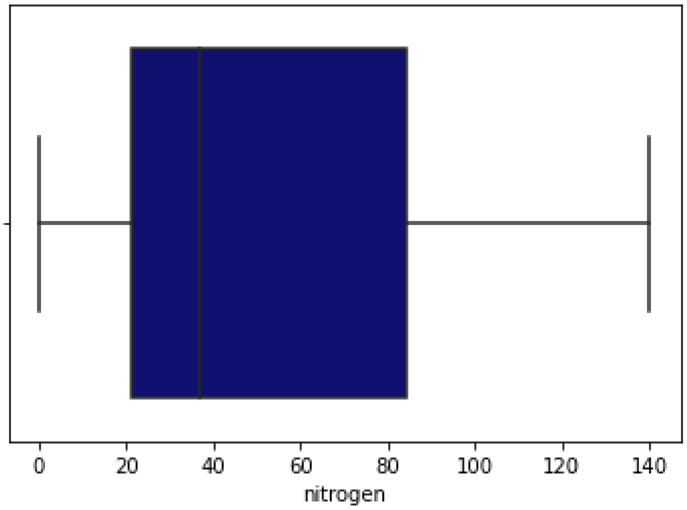
In [10]:
```
s.boxplot(df['phosphorus'], color='yellow')
```

Out[10]:
```
<AxesSubplot:xlabel='phosphorus'>
```



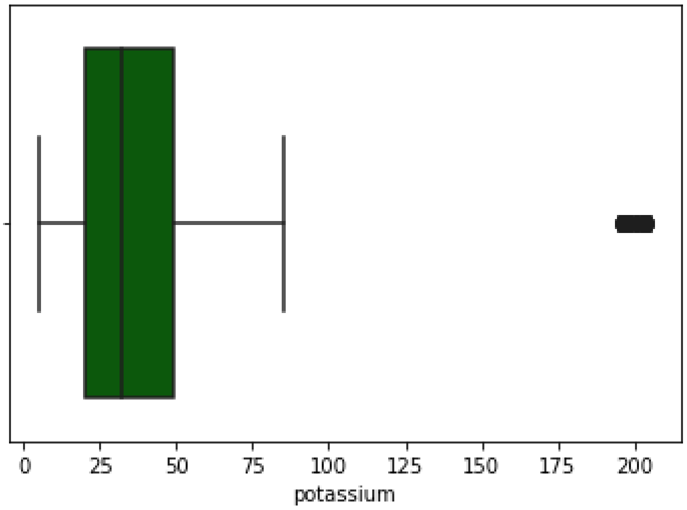
In [11]:
```
s.boxplot(df['nitrogen'], color='#000080')
```

```
Out[11]: <AxesSubplot:xlabel='nitrogen'>
```



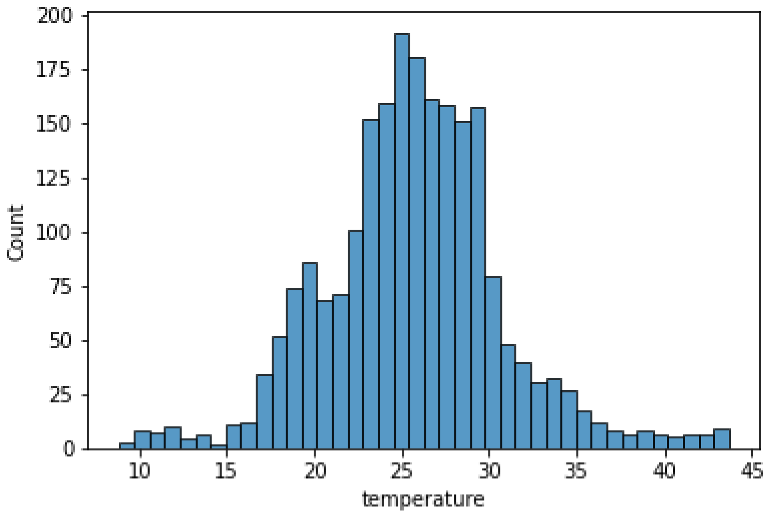
```
In [12]: s.boxplot(df['potassium'], color='darkgreen')
```

```
Out[12]: <AxesSubplot:xlabel='potassium'>
```



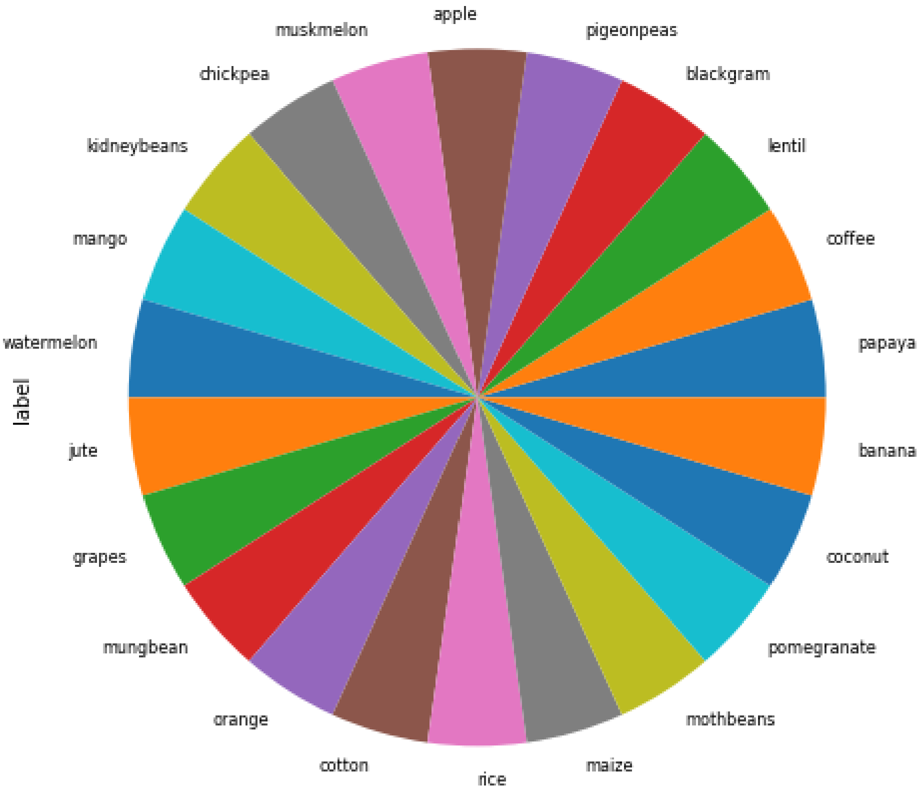
```
In [13]: s.histplot(df['temperature'])
```

```
Out[13]: <AxesSubplot:xlabel='temperature', ylabel='Count'>
```



```
In [14]:
def PropByVar(df,variable): #Propagation by variable
    pie_var = df[variable].value_counts()
    ax = pie_var.plot.pie(figsize=(8,8), fontsize =8)
    return "Different types of crops"

PropByVar(df,'label')
```

```
Out[14]: 'Different types of crops'
```


apple
pigeonpeas
blackgram
lentil
coffee
papaya
banana
coconut
pomegranate
mothbeans
maize
rice
cotton
orange
mungbean
grapes
jute
watermelon
mango
kidneybeans
chickpea
muskmelon
label