

# Análise das Características do Solo ideiais para cada Cultura

## Projeto da disciplina SME0860 - Aprendizado de Máquina Aplicado a Problemas

### Membros

- Alexandre E. de Souza Jesus - alexandre\_souza@usp.br - **12559506**
- Eduardo Zaffari Monteiro - eduardozaffarimonteiro@usp.br - **12559490**
- Lucas Ivars Cadima Ciziks - luciziks@usp.br - **12559472**

## 1. Introdução

O uso do solo de maneira adequada é de fundamental importância para maximizar o retorno do plantio e mantê-lo em boas condições para que possa ser utilizado por vários anos sem degradação. Caso o solo seja mal manejado, pode-se acabar com um terreno infértil, o que aumenta a demanda de recursos para o cultivo e recuperação da área de plantio.

Conforme o solo é utilizado para o plantio de um tipo de alimento, ocorre a diminuição dos tipos de nutrientes consumidos por essa planta, e em contrapartida os outros se tornam abundantes pelo acúmulo durante o tempo em que não foi consumido. Dessa forma, pode-se realizar a rotação de culturas, que consiste em plantar alimentos que consomem nutrientes diferentes de maneira cíclica, fazendo com que o solo se mantenha mais bem preservado.

Com objetivo de facilitar a escolha da cultura a ser semeada em um terreno específico será feito o treinamento de um modelo de aprendizado supervisionado. Para isso, serão utilizados dados que consistem em características do terreno, principalmente relacionadas à quantidade de nutrientes e substâncias nele presentes. A classificação dar-se-á de acordo com um tipo de cultura que é considerado ideal para o solo observado.

A tabela de dados apresenta 2200 diferentes condições de solo e suas respectivas culturas ideais, as quais são divididas em 22 classes que indicam diferentes sementes e frutas. Para cada um desses terrenos existem valores de quantidade de nitrogênio, fósforo, potássio, além de temperatura, umidade e pH, além da precipitação plantação recebe durante o crescimento.

## 2. Metodologia

O projeto está sendo realizado e versionado remotamente através da plataforma Github. Seu acesso é possível por meio do link <https://github.com/ale-souza/crop-recommendation>

### 2.1. Origem dos Dados

Os dados foram obtidos diretamente da plataforma kaggle, um site para estudo de ciência de dados e machine learning, e podem ser obtidos através do link <https://www.kaggle.com/datasets/aksahaha/crop-recommendation>. Segundo o usuário Abhishek Kumar, que disponibilizou os dados, eles são provenientes do ICAR (Indian Council of Agriculture Research), e complementados por pesquisas na internet feitas por ele.

## 2.2 Dicionário de Dados

- **Nitrogênio (nitrogen):** Representa a quantidade de nitrogênio (em kg/ha) presente no solo para a cultura. O nitrogênio é um nutriente essencial para o crescimento de plantas, e sua deficiência ou excesso pode afetar o crescimento e a produção da cultura;
- **Fósforo (phosphorus):** Representa a quantidade de fósforo (em kg/ha) presente no solo para a cultura. Também é um elemento essencial no plantio, sendo importante para processos como transferência de energia e fotossíntese;
- **Potássio (potassium):** Representa a quantidade de potássio (em kg/ha) presente no solo para a cultura. Também é um elemento essencial, e é importante para processos fisiológicos como regulação de água e transporte de nutrientes;
- **Temperatura (temperature):** Representa a temperatura média (em Celsius) durante o período de crescimento da cultura. A temperatura é um fator ambiental importante que pode afetar o crescimento e o desenvolvimento das plantas, e cada cultura possui uma temperatura ideal;
- **Umidade (humidity):** Representa a umidade relativa (em porcentagem) durante o período de crescimento da cultura. A umidade é outro fator ambiental importante, tendo em vista que uma alta umidade pode promover a proliferação de fungos e desenvolvimento de doenças;
- **pH:** Representa o pH da cultura durante seu período de crescimento. O pH é uma medida de acidez ou alcalinidade do solo e pode afetar a disponibilidade de nutrientes para a cultura;
- **Precipitação (rainfall):** Representa a precipitação (em mm) durante o período de crescimento da cultura. Cada cultura necessita de uma quantidade diferente de água, o que torna a precipitação outro fator ambiental importante;
- **Crop (label):** Representa o tipo da cultura.

## 3. Coleta dos Dados

```
In [10]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
```

```
In [11]: # Leitura dos dados do problema
df = pd.read_csv('https://raw.githubusercontent.com/ale-souza/crop-recommendation/main
Crop_recommendation.csv?token=GHSAT0AAAAACB3RUJBLKCECF2HDREQ7OM2ZCQHG
df.head()
```

```
Out[11]: Nitrogen  phosphorus  potassium  temperature  humidity  ph  rainfall  label  Unnamed:  Unnamed:
```

									8	9
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	NaN	NaN
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	NaN	NaN
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	NaN	NaN
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	NaN	NaN
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	NaN	NaN

```
In [12]: # Limpando dataset
df = df.drop(['Unnamed: 8', 'Unnamed: 9'], axis=1)
df = df.rename(columns={"label": "crop"})
df.head()
```

```
Out[12]:
```

	Nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall	crop
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

```
In [13]: # Verificando as categorias de plantação
labels = df['crop'].astype('category').values
labels = list(labels.categories)
labels
```

```
Out[13]: ['apple',
'banana',
'blackgram',
'chickpea',
'coconut',
'coffee',
'cotton',
'grapes',
'jute',
'kidneybeans',
'lentil',
'maize',
'mango',
'mothbeans',
'mungbean',
'muskmelon',
'orange',
'papaya',
'pigeonpeas',
'pomegranate',
'rice',
'watermelon']
```

## 4. Análise Exploratória dos Dados

### 4.1. Medidas Descritivas

```
In [14]: # Função para calcular o coeficiente de variância (CV)
def coeficiente_variancia(table):
    return 100 * table.std() / table.mean()

# Função para calcular a amplitude
def amplitude(table):
    return table.max() - table.min()

# Aplicando medidas de posição e dispersão aos atributos preditivos
medidas_descritivas = df.drop(["crop"], axis=1).agg(["min", "max", "mean", "median", "va
```

```
In [15]: # Renomeando das medidas descritivas
novos_nomes = {
    "min": "Minimo",
    "max": "Maximo",
    "mean": "Media",
    "median": "Mediana",
    "var": "Variancia",
    "std": "Desvio-padrao",
    "coeficiente_variancia": "Coeficiente de Variancia",
    "amplitude": "Amplitude"
}

medidas_descritivas = medidas_descritivas.rename(novos_nomes)

# Arredondando casas decimais das medidas descritivas e de dispersão
medidas_descritivas = medidas_descritivas.round(3)
```

```
In [16]: medidas_descritivas
```

```
Out[16]:
```

	Nitrogen	phosphorus	potassium	temperature	humidity	ph	rainfall
<b>Minimo</b>	0.000	5.000	5.000	8.826	14.258	3.505	20.211
<b>Maximo</b>	140.000	145.000	205.000	43.675	99.982	9.935	298.560
<b>Media</b>	50.552	53.363	48.149	25.616	71.482	6.469	103.464
<b>Mediana</b>	37.000	51.000	32.000	25.599	80.473	6.425	94.868
<b>Variancia</b>	1362.890	1088.068	2565.213	25.642	495.677	0.599	3020.424
<b>Desvio-padrao</b>	36.917	32.986	50.648	5.064	22.264	0.774	54.958
<b>Coeficiente de Variancia</b>	73.029	61.814	105.190	19.768	31.146	11.963	53.119
<b>Amplitude</b>	140.000	140.000	200.000	34.850	85.724	6.430	278.349

A partir das medidas descritivas, pode-se ter uma ideia inicial das distribuições das características. É possível inferir que as variáveis *temperature* e *pH* possuem uma curva simétrica, já que suas médias e medianas são bem próximas, enquanto *humidity* provavelmente possui uma curva assimétrica à esquerda. Para todos os outros atributos as curvas são possivelmente assimétricas à direita.

No que tange as medidas de dispersão, a análise anterior é reforçada. As medidas de variância e desvio-padrão apresentam valores altos para as variáveis que não são simétricas, o que indica que há uma alta variabilidade nos dados. Ou seja, há valores que possuem uma grande distância da média.

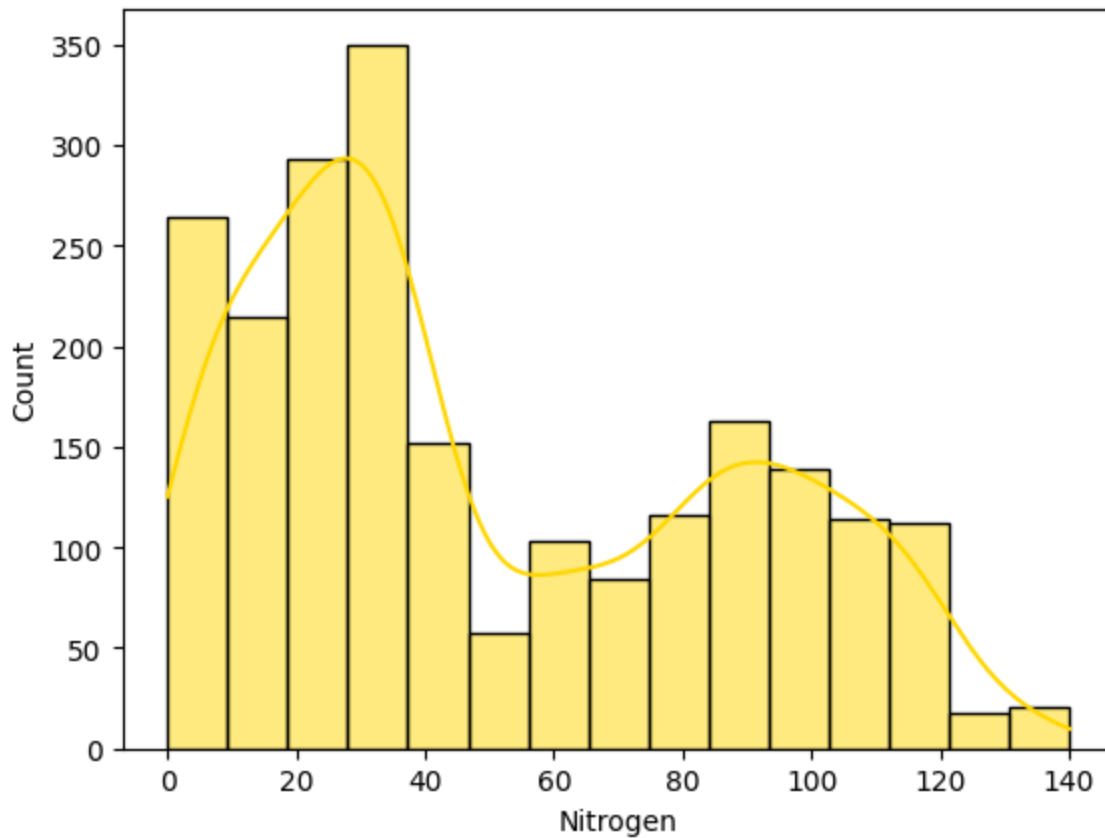
## 4.2. Visualização dos Dados

### 4.2.1. Histogramas

Abaixo, é possível observar o formato da curva dos atributos. Assim, há ainda mais evidências de que a análise anteriormente feita está, provavelmente, correta. Fazem-se necessários, então, testes de hipóteses.

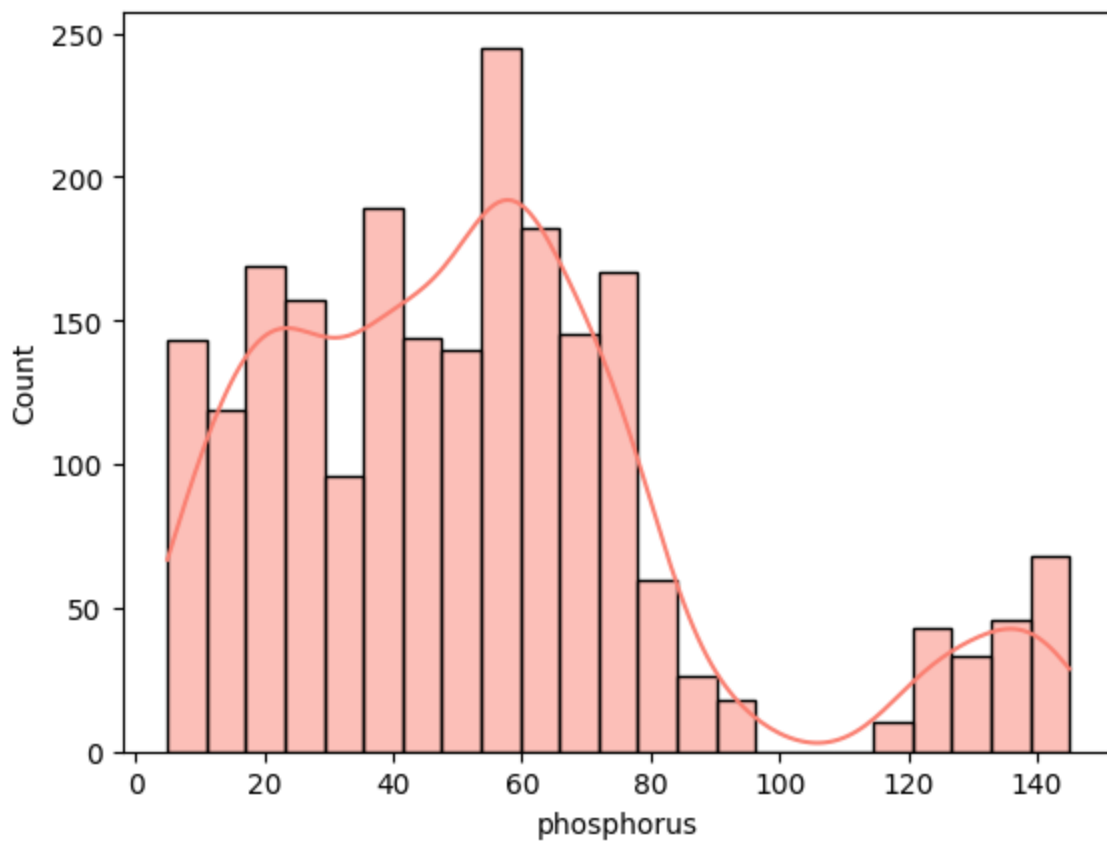
```
In [17]: # Nitrogen
sns.histplot(data=df, x="Nitrogen", kde=True, color="gold")
```

```
Out[17]: <Axes: xlabel='Nitrogen', ylabel='Count'>
```



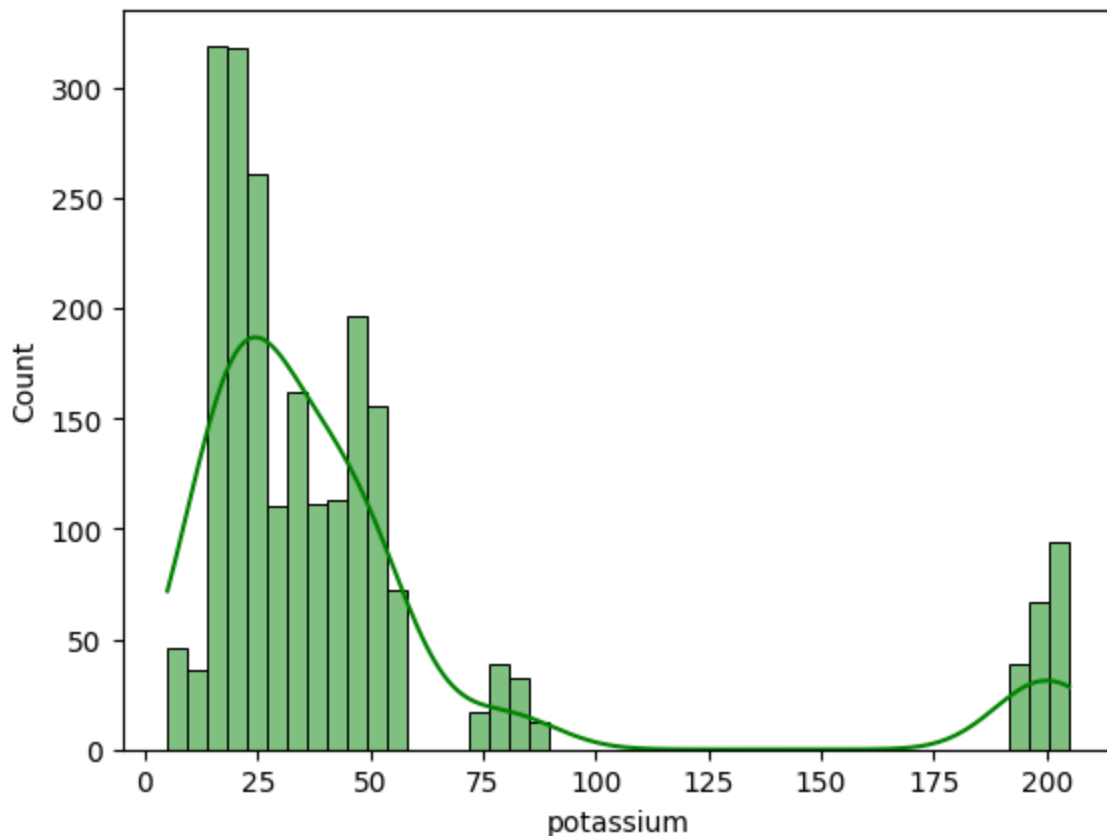
```
In [18]: # Phosphorus
sns.histplot(data=df, x="phosphorus", kde=True, color="salmon")
```

```
Out[18]: <Axes: xlabel='phosphorus', ylabel='Count'>
```



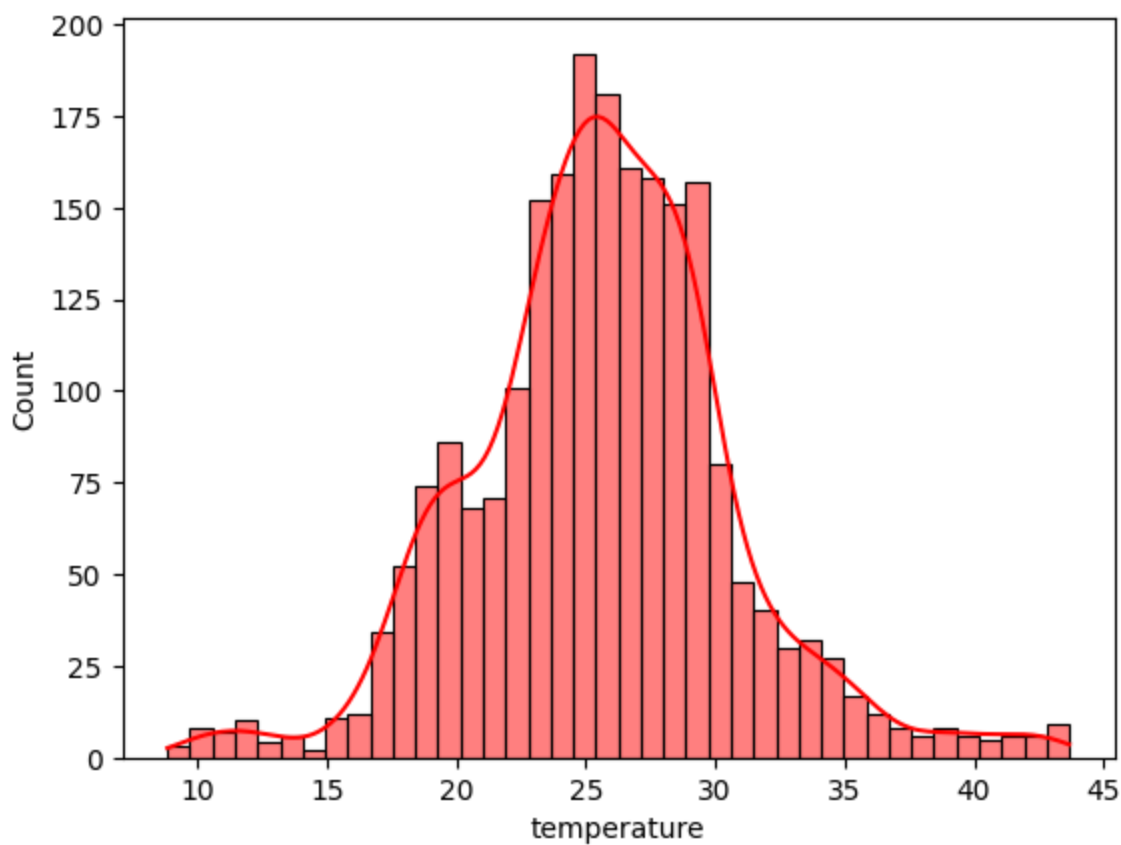
```
In [19]: # Potassium
sns.histplot(data=df, x="potassium", kde=True, color="green")
```

```
Out[19]: <Axes: xlabel='potassium', ylabel='Count'>
```



```
In [20]: # temperature
sns.histplot(data=df, x="temperature", kde=True, color="red")
```

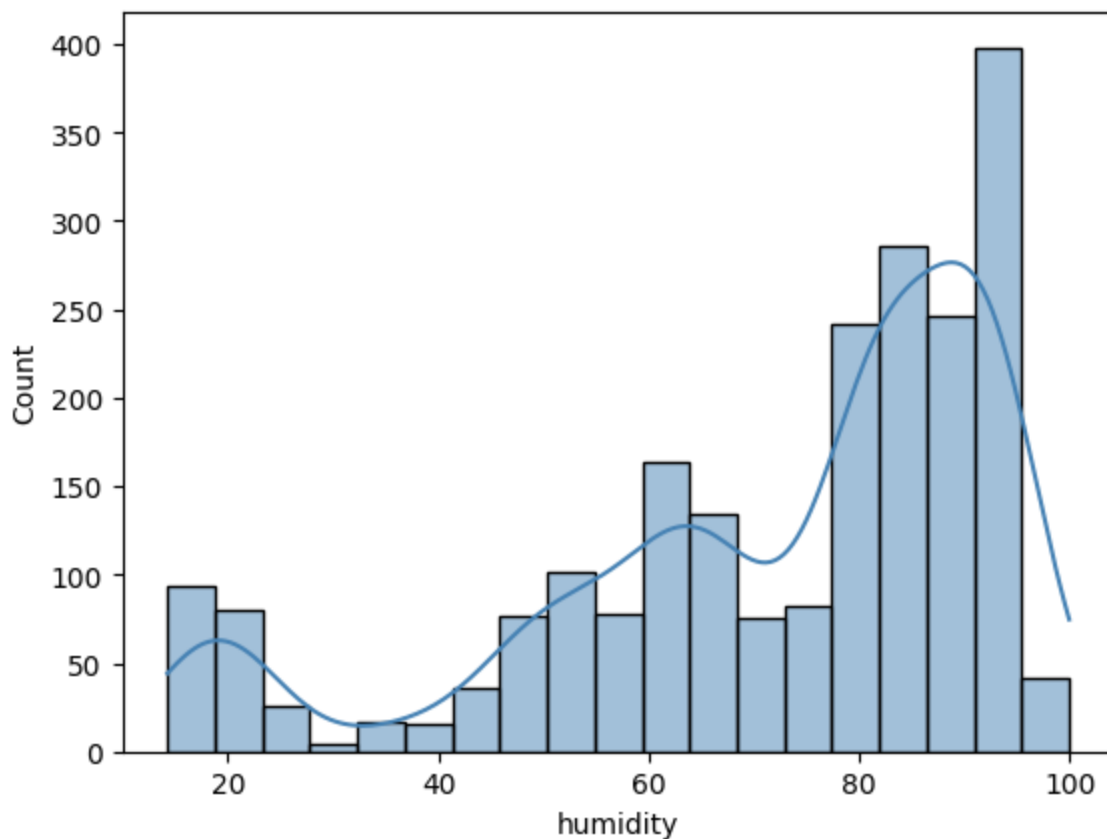
```
Out[20]: <Axes: xlabel='temperature', ylabel='Count'>
```



Percebe-se, como anteriormente dito, que a distribuição dos dados referentes à variável temperatura provavelmente segue uma distribuição.

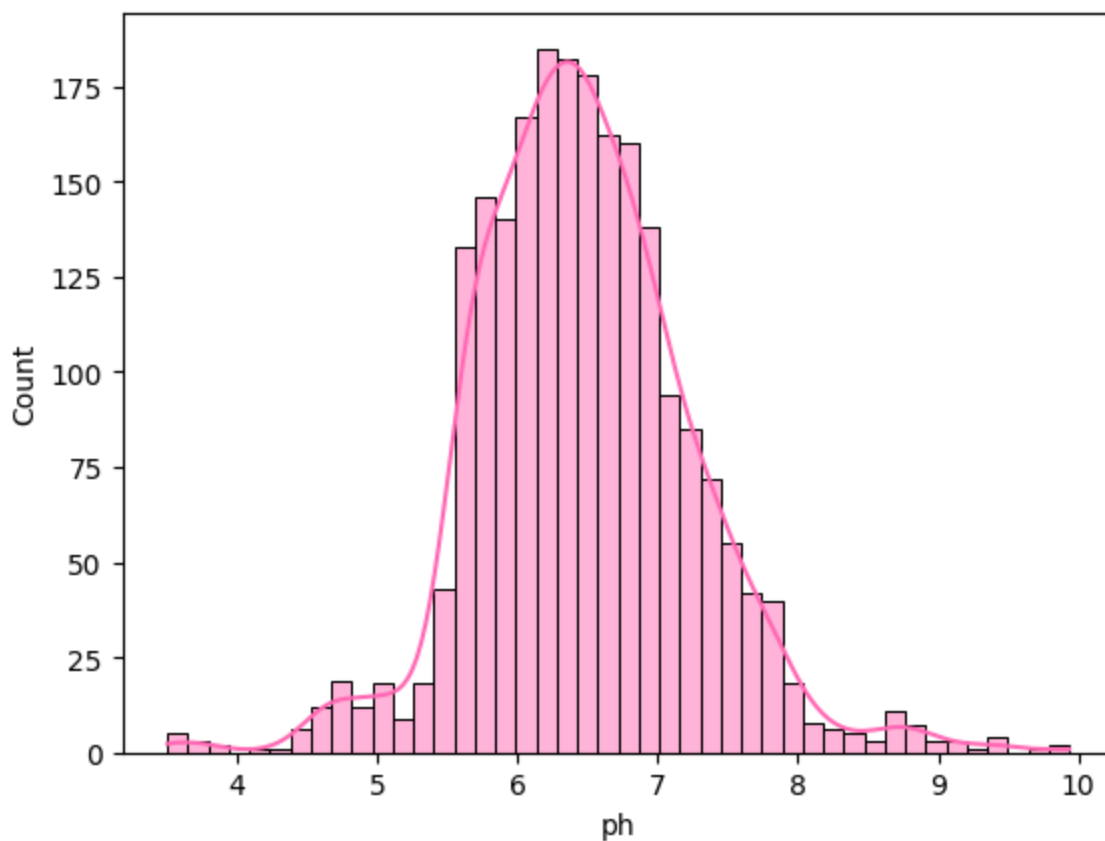
```
In [21]: # Humidity
sns.histplot(data=df, x="humidity", kde=True, color="steelblue")
```

```
Out[21]: <Axes: xlabel='humidity', ylabel='Count'>
```



```
In [22]: # ph
sns.histplot(data=df, x="ph", kde=True, color="hotpink")
```

```
Out[22]: <Axes: xlabel='ph', ylabel='Count'>
```

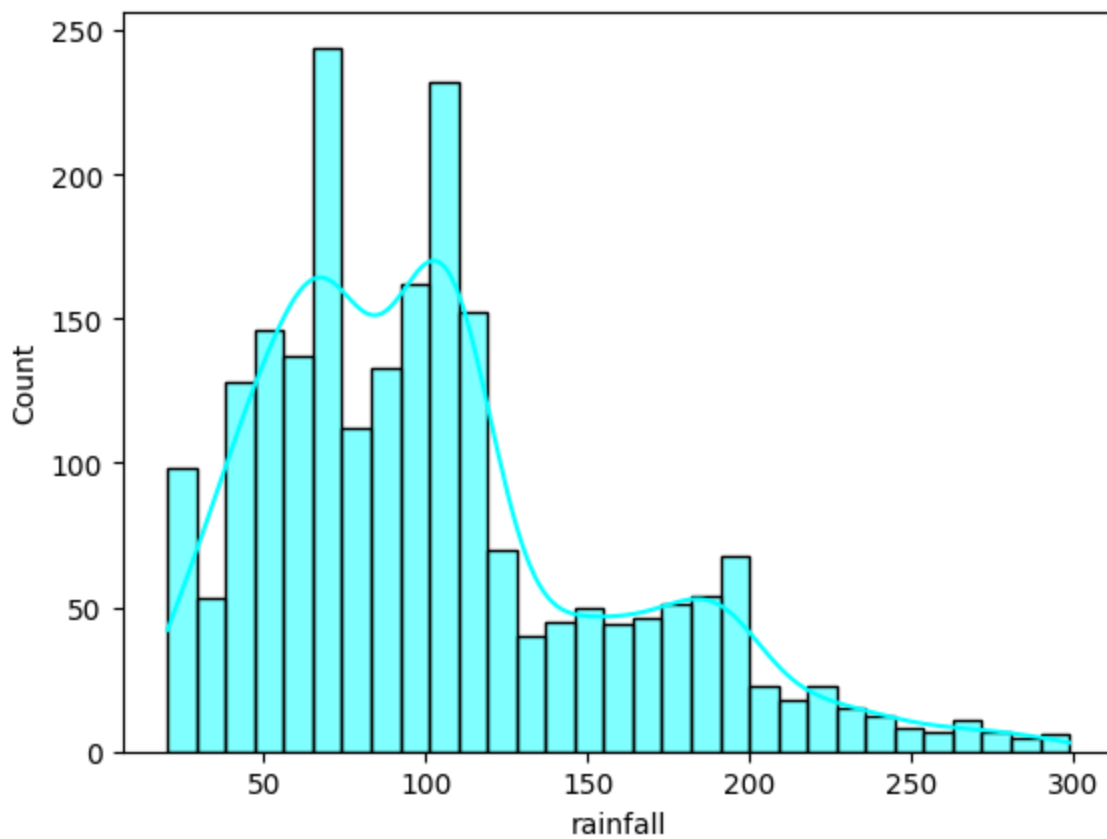


Percebe-se, como anteriormente dito, que a distribuição dos dados referentes à variável pH provavelmente segue uma distribuição.

```
In [23]: ## Rainfall
sns.histplot(data=df, x="rainfall", kde=True, color="aqua")
```

```
Out[23]: <Axes: xlabel='rainfall', ylabel='Count'>
```



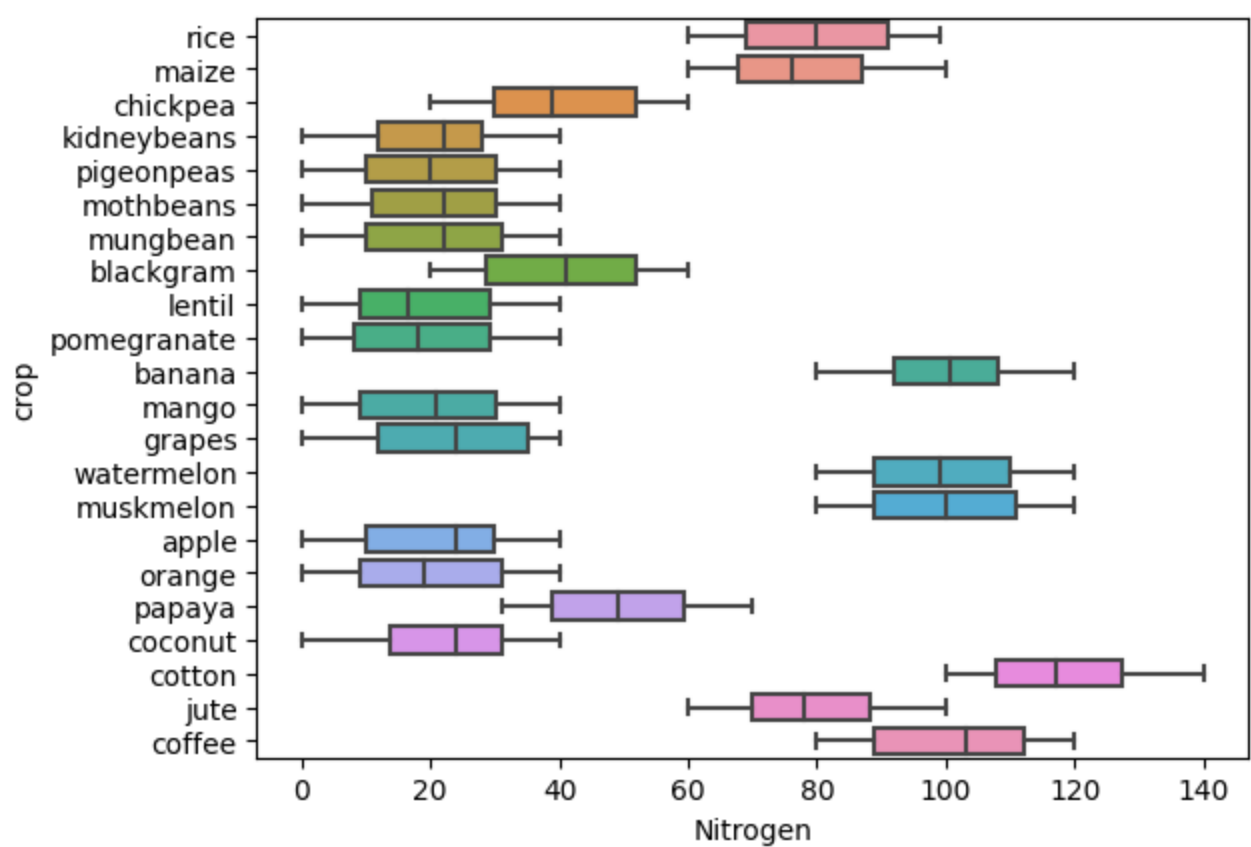


### 4.2.2. Boxplots

Com os *boxplots*, é possível comparar a distribuição dos dados em relação ao atributo-alvo. Mais uma vez, a teoria de que as variáveis *pH* e *temperature* são mais balanceadas é corroborada.

```
In [24]: # -----  
#   Nitrogen x Crop  
# -----  
  
sns.boxplot(x=df["Nitrogen"], y=df["crop"])
```

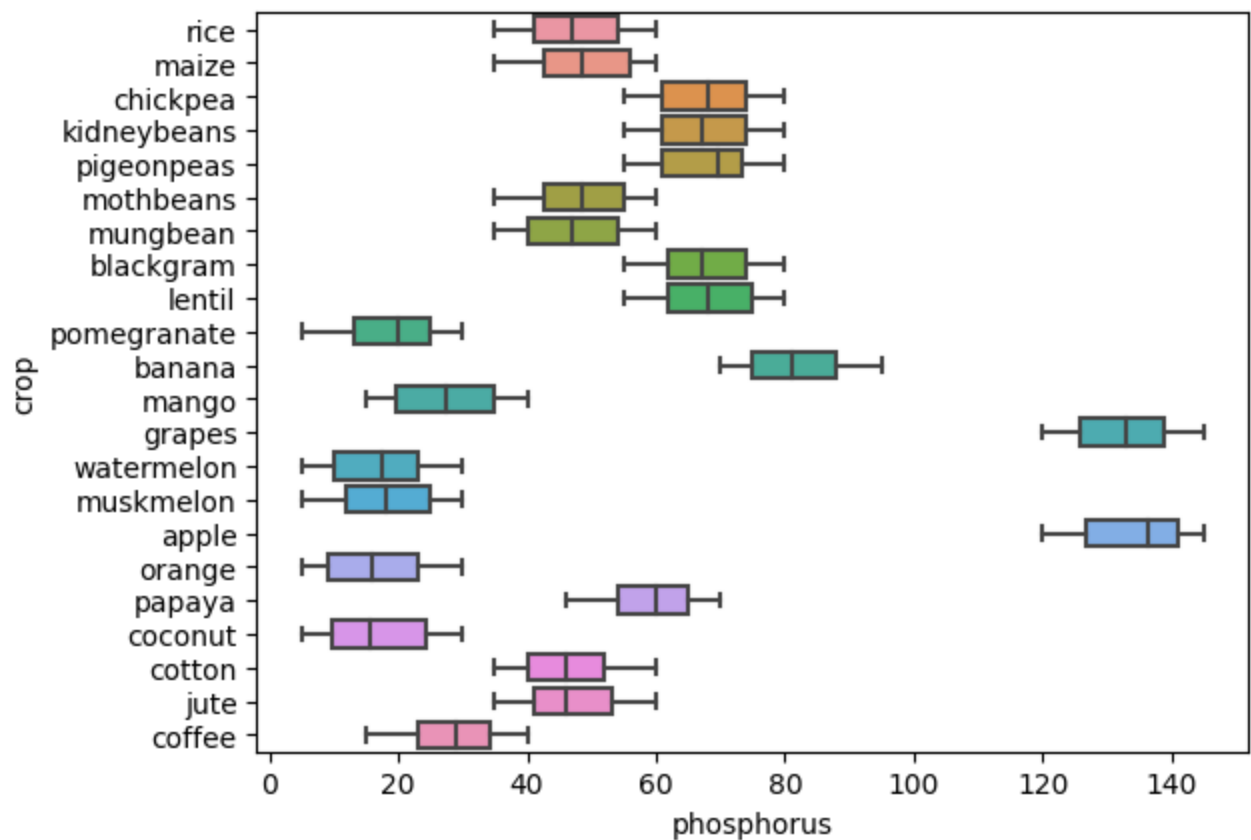
```
Out[24]: <Axes: xlabel='Nitrogen', ylabel='crop'>
```



```
In [25]: # -----
#   Phosphorus x Crop
#   -----

sns.boxplot(x=df["phosphorus"], y=df["crop"])
```

```
Out[25]: <Axes: xlabel='phosphorus', ylabel='crop'>
```

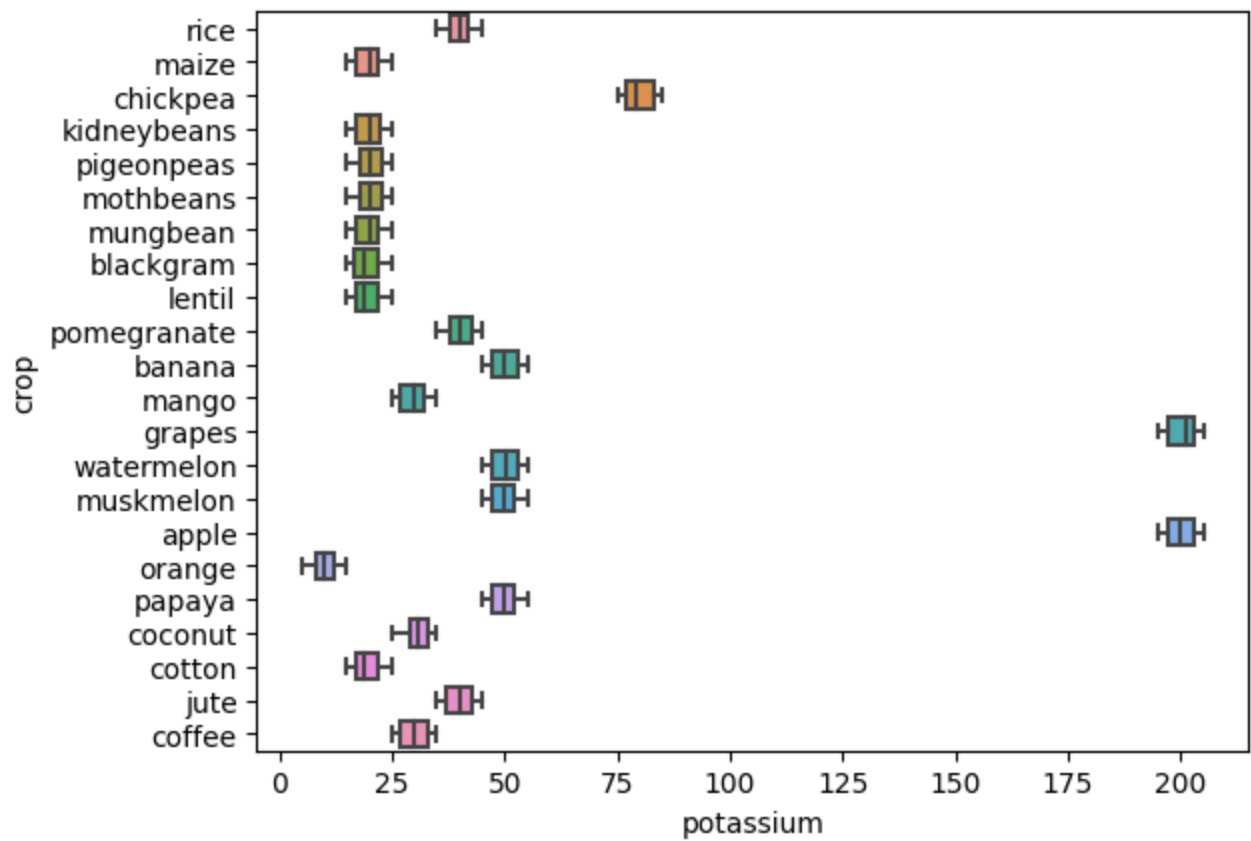


```
In [26]: # -----
```

```
# Potassium x Crop
# -----

sns.boxplot(x=df["potassium"], y=df["crop"])
```

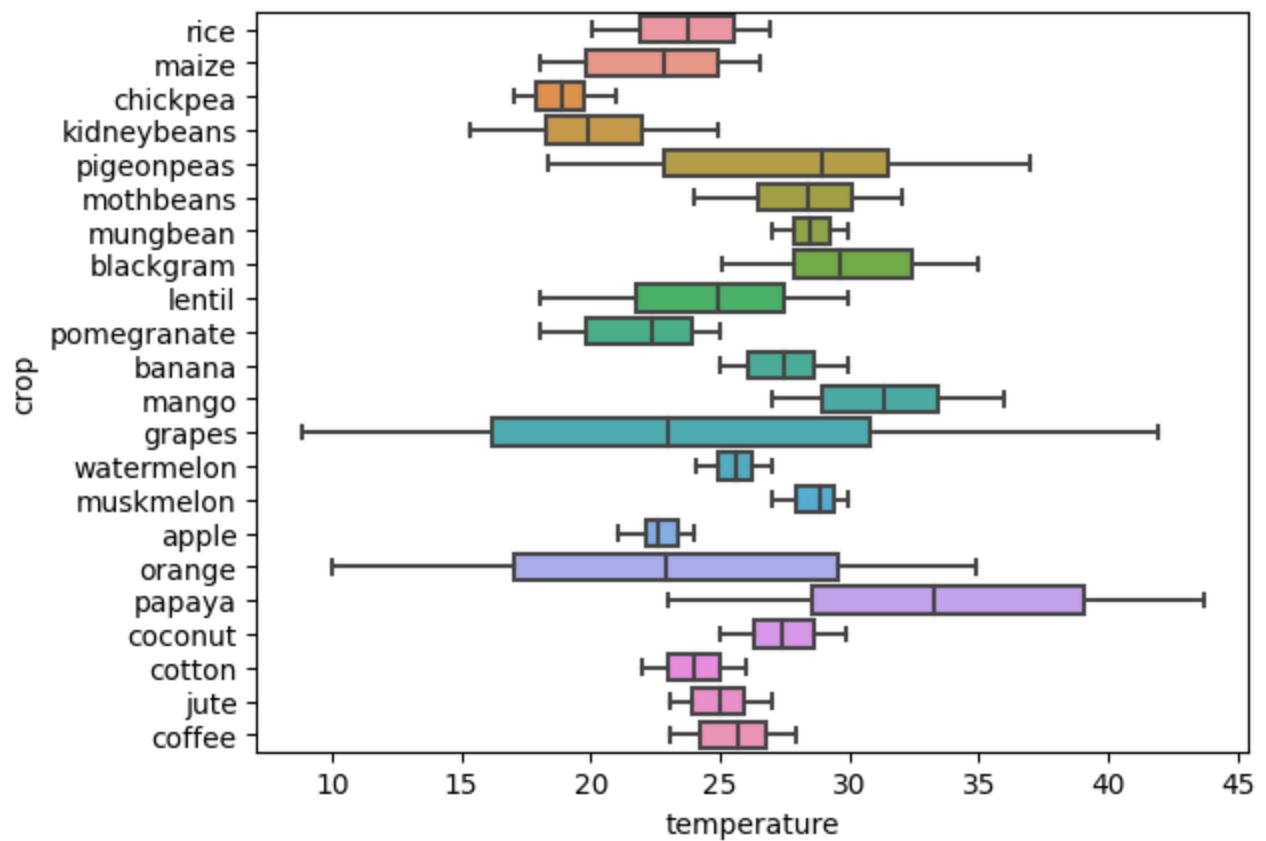
Out[26]: <Axes: xlabel='potassium', ylabel='crop'>



```
In [27]: # -----
# Temperature x Crop
# -----

sns.boxplot(x=df["temperature"], y=df["crop"])
```

Out[27]: <Axes: xlabel='temperature', ylabel='crop'>

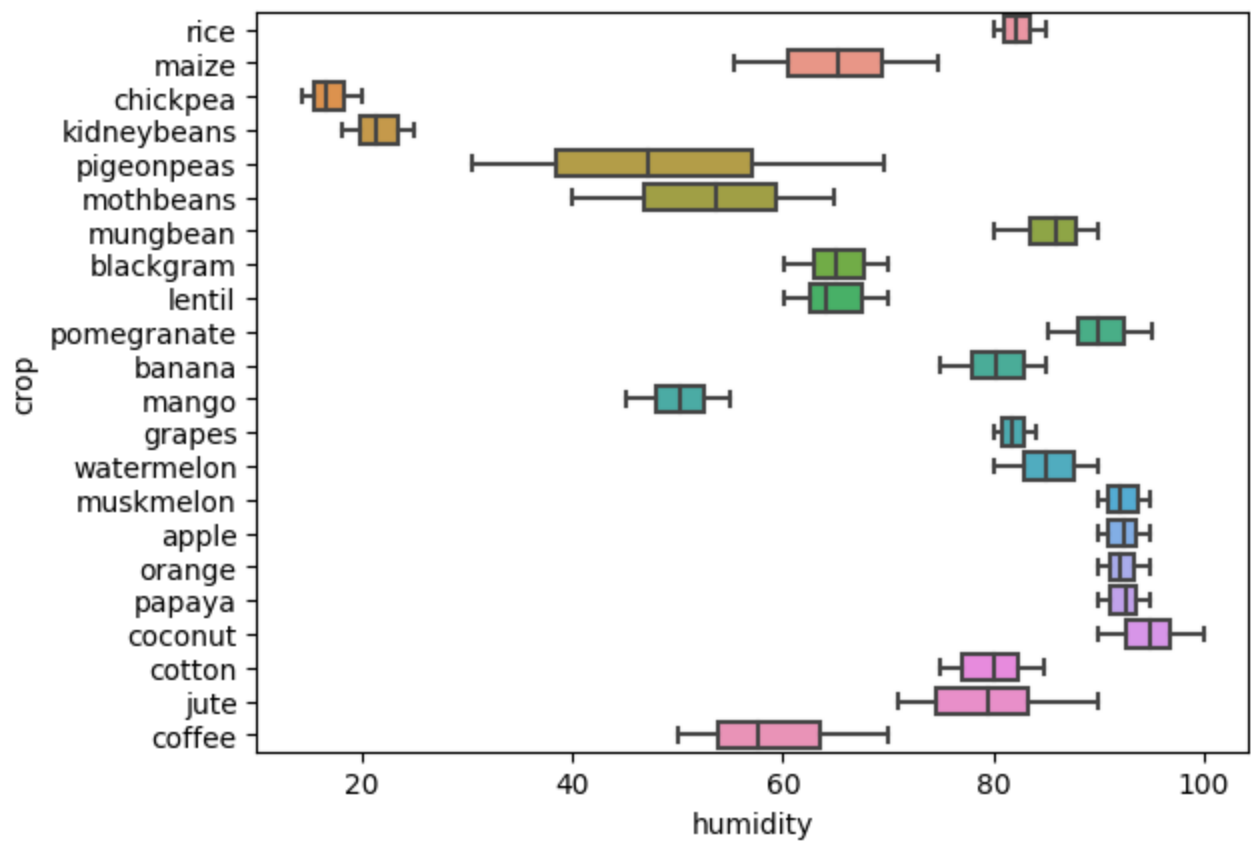


A maior parte dos valores está ao redor da média, que é de aproximadamente 25. Apesar de certos valores apresentarem grande variação, como *grapes* e *orange*, isso não afetou a curva.

```
In [28]: # -----
# Humidity x Crop
# -----

sns.boxplot(x=df["humidity"], y=df["crop"])
```

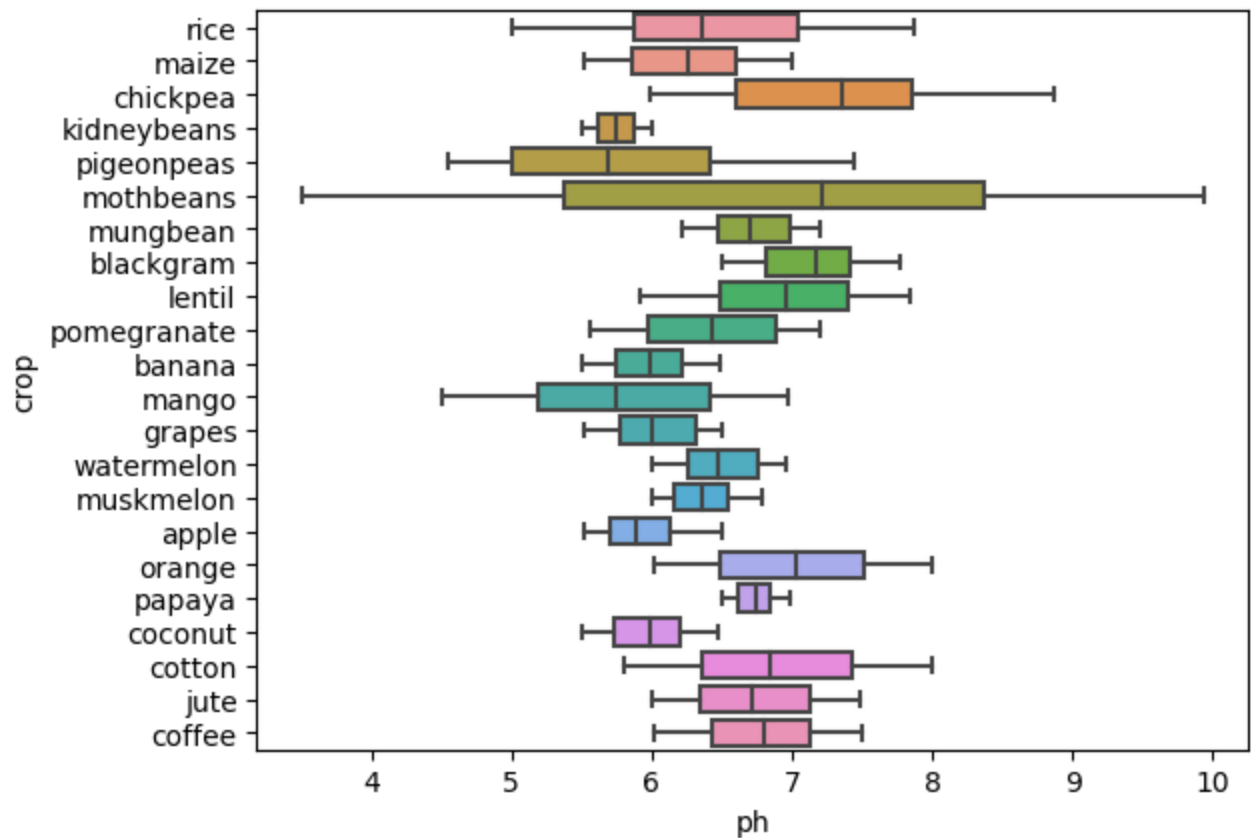
```
Out[28]: <Axes: xlabel='humidity', ylabel='crop'>
```



```
In [29]: # -----
#   ph x Crop
#   -----

sns.boxplot(x=df["ph"], y=df["crop"])
```

```
Out[29]: <Axes: xlabel='ph', ylabel='crop'>
```

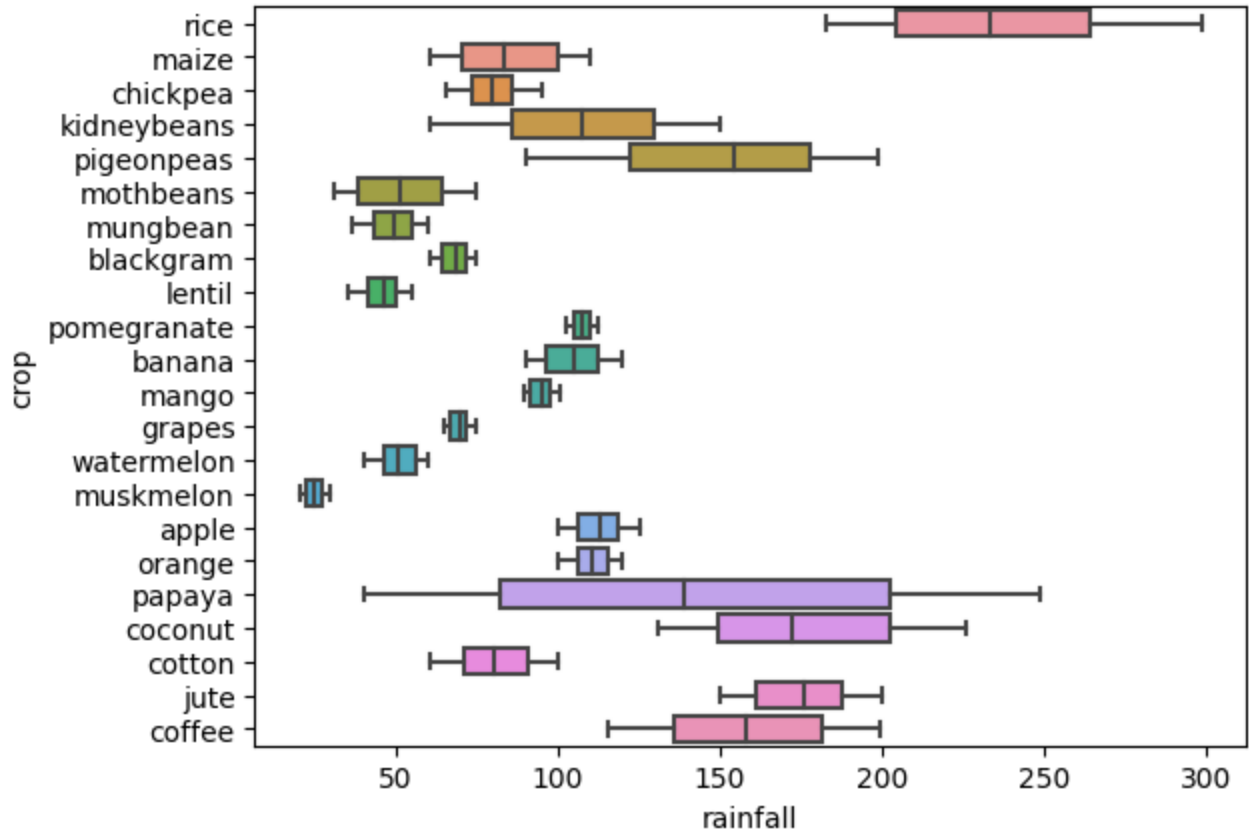


O padrão se repete com o atributo *pH*. A maior parte dos valores está concentrada ao redor da média.

Neste caso, tal resultado é previsível, visto que essa medida varia entre 0 e 14, e 7 representa um meio neutro (a média dos valores foi de aproximadamente 6,4)

```
In [30]: # -----  
# Rainfall x Crop  
# -----  
  
sns.boxplot(x=df["rainfall"], y=df["crop"])
```

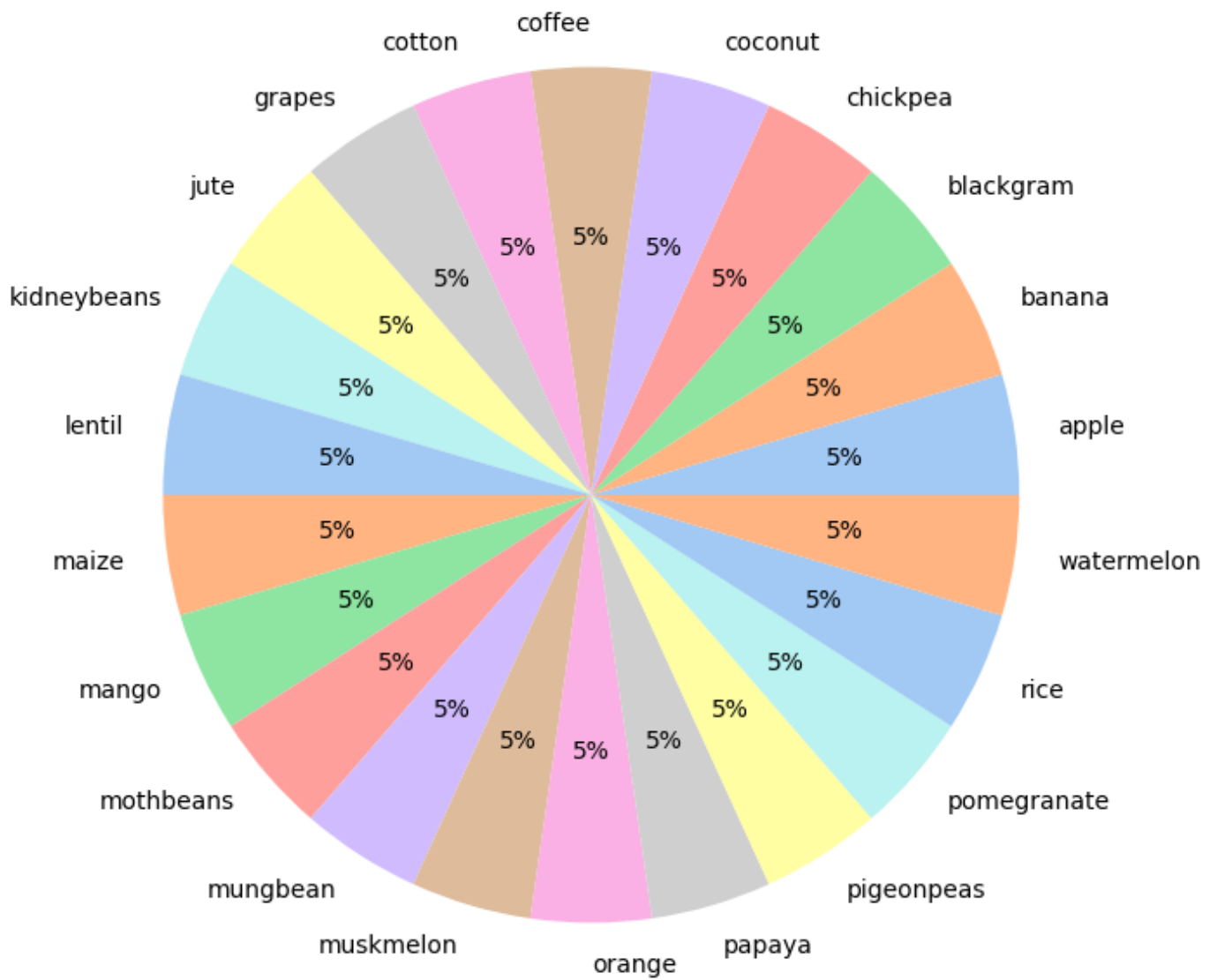
```
Out[30]: <Axes: xlabel='rainfall', ylabel='crop'>
```



É possível perceber, então, que há certa separação no que tange aos atributos para cada tipo de cultura. Há indícios, portanto, de que é possível classificar o rótulo de novas observações a partir deste conjunto de variáveis.

### 4.2.3. Balaceamento dos Dados

```
In [31]: plt.figure(figsize=(8, 8))  
colors = sns.color_palette('pastel')  
  
plt.pie(df['crop'].groupby(df['crop']).count(), labels = labels, colors = colors, autopc  
plt.show()
```

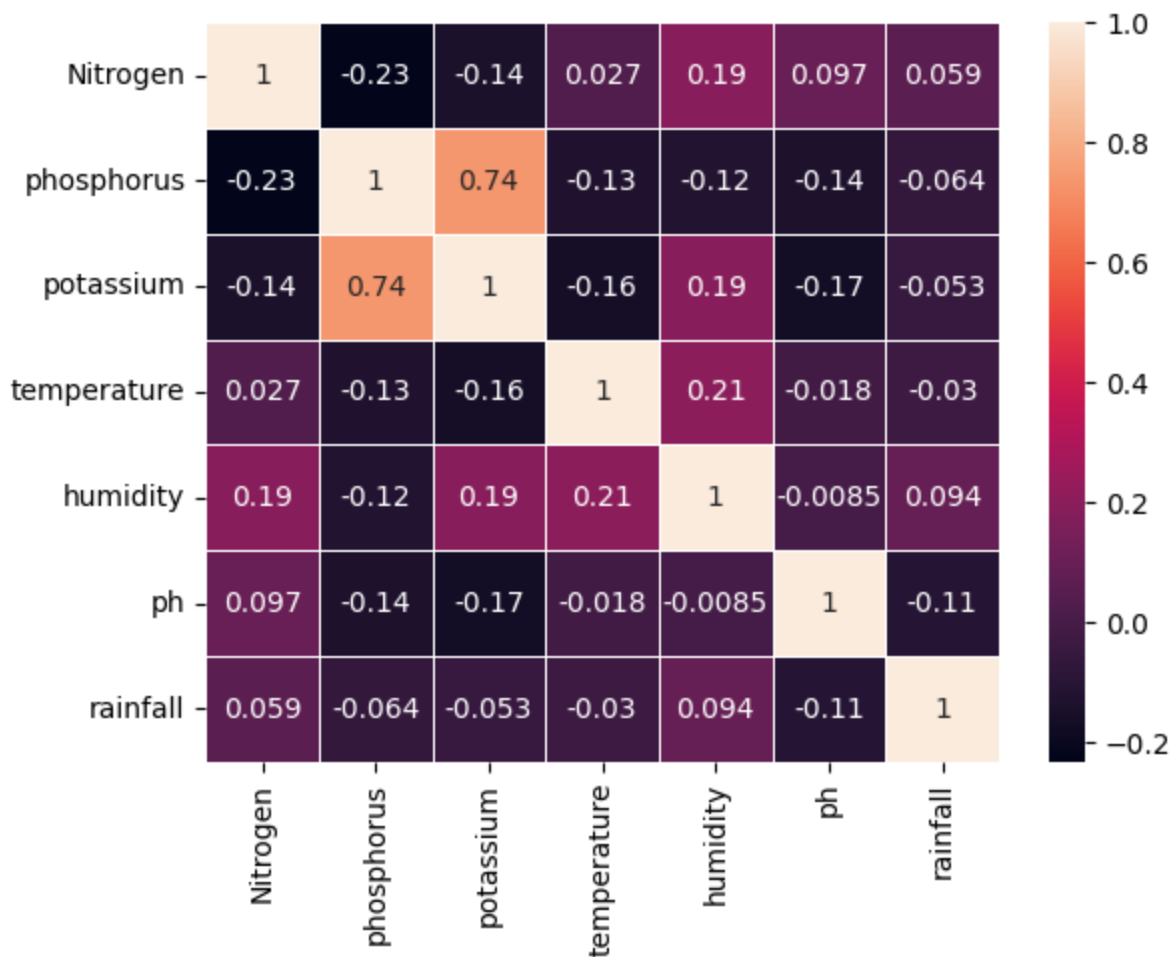


É possível perceber que a distribuição dos valores é idêntica.

## 4.3. Correlação entre as Variáveis

```
In [32]: # Mapa de calor com as correlações entre os atributos
sns.heatmap(df.drop(columns=["crop"]).corr(), annot=True, linewidths=0.5)
```

```
Out[32]: <Axes: >
```



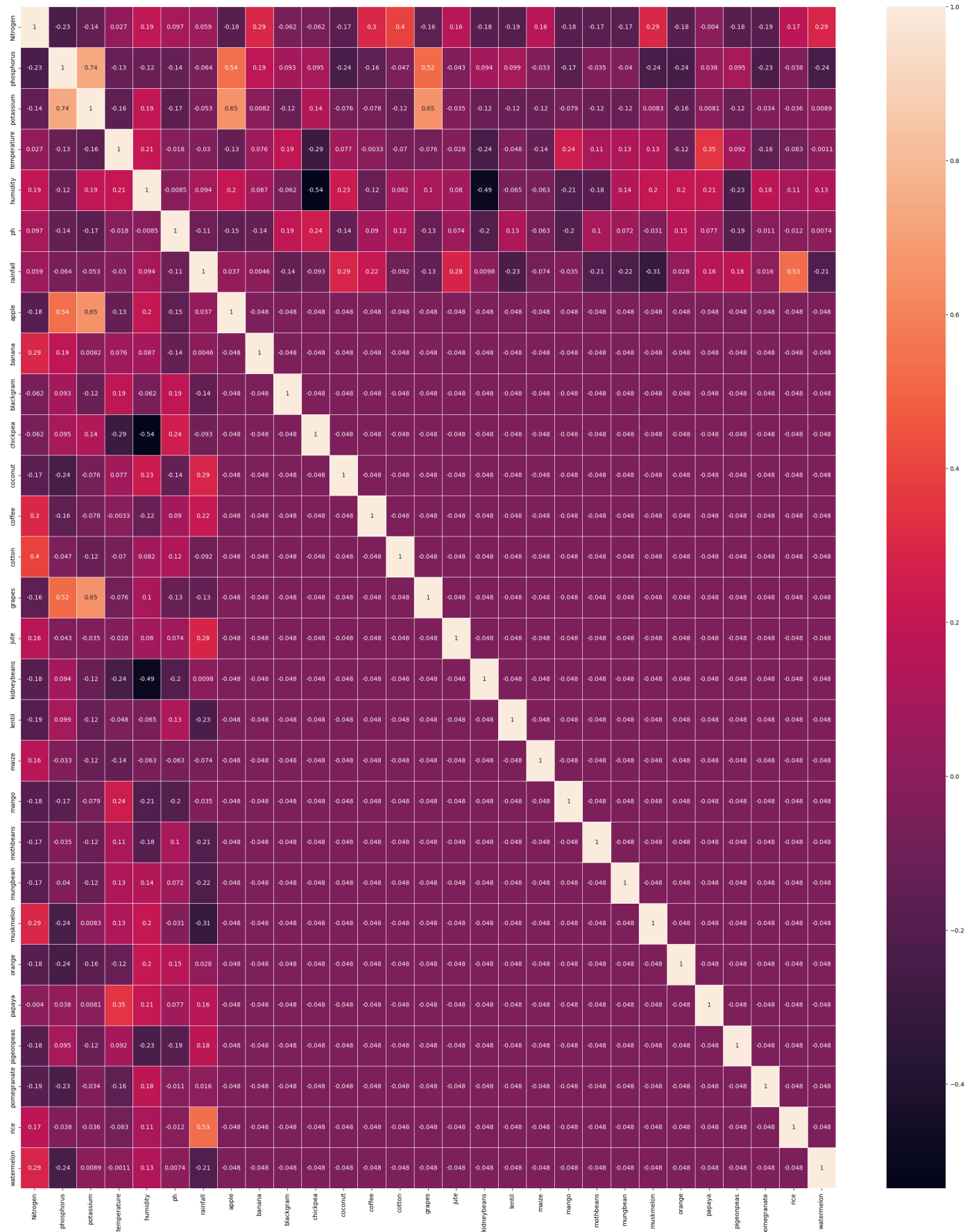
A maior parte das variáveis **não** está relacionada entre si, com exceção dos atributos *potassium* e *phosphorus*, que possuem uma correlação positiva considerável. Para que a classificação dos dados seja mais efetiva, um dos atributos pode ser removido. Abaixo é possível visualizar a **matriz de dispersão** entre todos as culturas possíveis (os rótulos) e os atributos, o que é necessário para avaliar qual das duas variáveis relacionadas poderia ser removida.

```
In [33]: dummies = pd.get_dummies(df['crop'])
dummy_df = pd.concat([df, dummies], axis=1)

plt.figure(figsize=(30, 35))
corr = dummy_df.drop(columns=["crop"]).corr()
sns.heatmap(corr, annot=True, linewidths=0.5)
```

Out[33]: <Axes: >

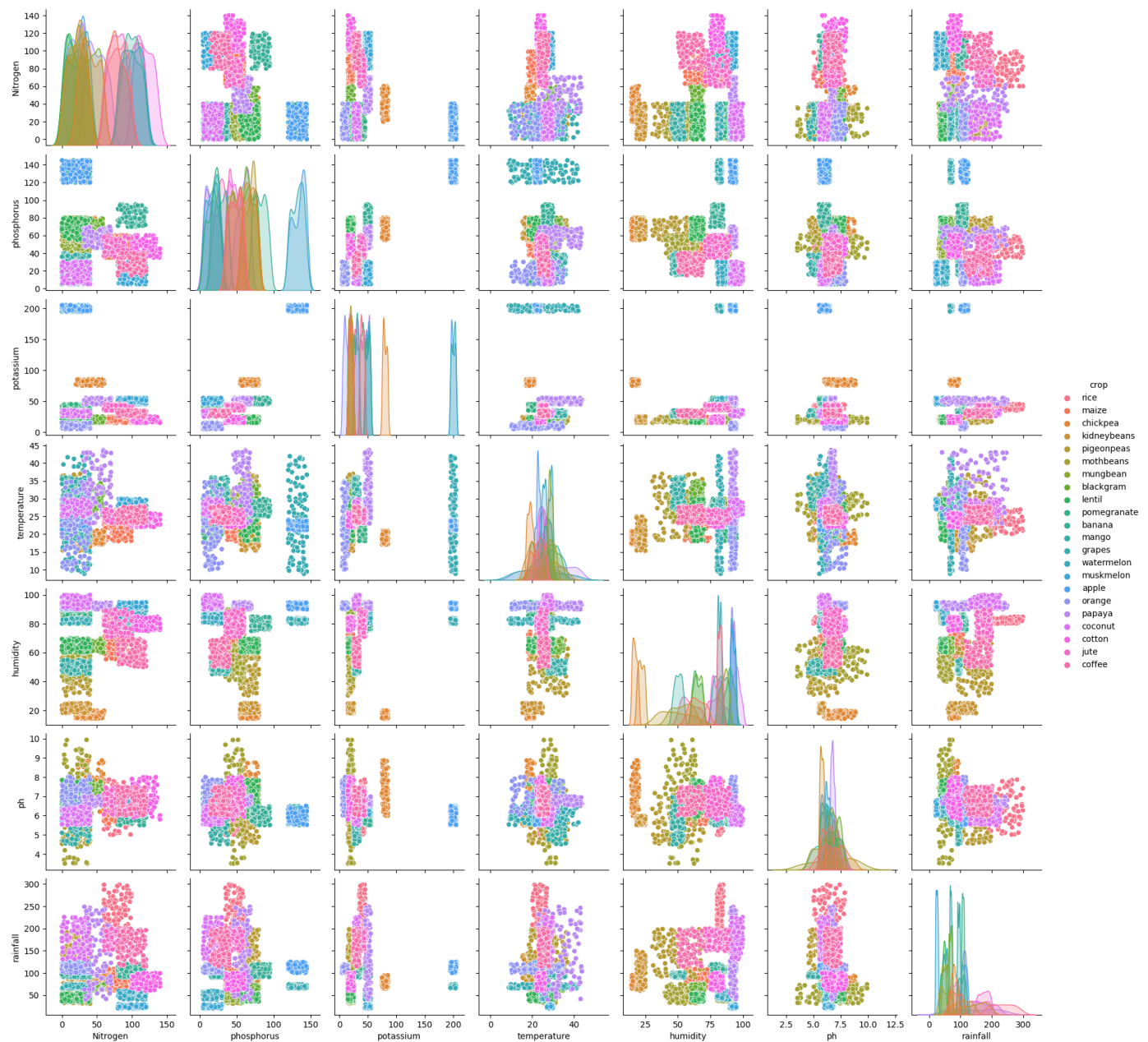




É possível perceber que dentre elas a que possui maior relação com os atributos-alvo é a *potassium*. Sendo assim, uma possível saída para o problema seria remover a variável *phosphorus* do conjunto de dados, visto que *potassium* tem maior relação com o atributo-alvo.

```
In [34]: # Interação entre os atributos preditivos
sns.pairplot(df, hue='crop')

<seaborn.axisgrid.PairGrid at 0x7f2d1a4967d0>
```



## 5. Treinamento dos Modelos

## 6. Avaliação dos Modelos

## 7. Referências Bibliográficas

- SISTEMA DE PRODUÇÃO DE MELÃO. [S. I.]: Embrapa Semiárido, ISSN 1807-0027. Mensal. Disponível em: [http://www.cpatsa.embrapa.br:8080/sistema\\_producao/spmelao/manejo\\_do\\_solo.html](http://www.cpatsa.embrapa.br:8080/sistema_producao/spmelao/manejo_do_solo.html). Acesso em: 1 maio 2023;
- SOUSA, Rafaela. Rotação de culturas. [S. I.]: Brasil Escola. Disponível em: <https://brasilecola.uol.com.br/geografia/rotacao-culturas.htm>. Acesso em: 1 maio 2023.

```
In [2]: !jupyter nbconvert --to webpdf --allow-chromium-download "Análise de Culturas.ipynb"
```

```
[NbConvertApp] Converting notebook Análise de Culturas.ipynb to webpdf
```

```
[NbConvertApp] Building PDF
```

```
[INFO] Starting Chromium download.
```

```
0%|          | 0.00/137M [00:00<?, ?b/s]
0%|          | 123k/137M [00:00<01:53, 1.21Mb/s]
1%|          | 809k/137M [00:00<00:30, 4.49Mb/s]
1%|1         | 1.79M/137M [00:00<00:19, 6.88Mb/s]
2%|2         | 2.83M/137M [00:00<00:16, 8.18Mb/s]
3%|2         | 3.93M/137M [00:00<00:14, 9.19Mb/s]
4%|3         | 4.99M/137M [00:00<00:13, 9.64Mb/s]
4%|4         | 6.11M/137M [00:00<00:12, 10.1Mb/s]
5%|5         | 7.25M/137M [00:00<00:12, 10.5Mb/s]
6%|6         | 8.38M/137M [00:00<00:12, 10.7Mb/s]
7%|6         | 9.45M/137M [00:01<00:11, 10.7Mb/s]
8%|7         | 10.6M/137M [00:01<00:11, 10.8Mb/s]
9%|8         | 11.7M/137M [00:01<00:12, 10.2Mb/s]
9%|9         | 12.7M/137M [00:01<00:11, 10.4Mb/s]
10%|#        | 13.9M/137M [00:01<00:11, 10.7Mb/s]
11%|#        | 15.0M/137M [00:02<00:33, 3.63Mb/s]
12%|#1       | 16.1M/137M [00:02<00:26, 4.56Mb/s]
13%|#2       | 17.2M/137M [00:02<00:21, 5.55Mb/s]
13%|#3       | 18.3M/137M [00:02<00:18, 6.49Mb/s]
14%|#4       | 19.5M/137M [00:02<00:15, 7.51Mb/s]
15%|#5       | 20.6M/137M [00:02<00:13, 8.37Mb/s]
16%|#5       | 21.8M/137M [00:03<00:36, 3.12Mb/s]
17%|#6       | 22.9M/137M [00:03<00:29, 3.91Mb/s]
18%|#7       | 24.0M/137M [00:03<00:23, 4.89Mb/s]
18%|#8       | 25.0M/137M [00:03<00:20, 5.49Mb/s]
19%|#9       | 26.0M/137M [00:04<00:17, 6.39Mb/s]
20%|#9       | 27.2M/137M [00:04<00:14, 7.37Mb/s]
21%|##       | 28.3M/137M [00:04<00:13, 8.22Mb/s]
21%|##1      | 29.4M/137M [00:04<00:12, 8.89Mb/s]
22%|##2      | 30.5M/137M [00:04<00:11, 9.38Mb/s]
23%|##3      | 31.5M/137M [00:04<00:10, 9.73Mb/s]
24%|##3      | 32.7M/137M [00:04<00:10, 10.1Mb/s]
25%|##4      | 33.7M/137M [00:04<00:10, 10.1Mb/s]
25%|##5      | 34.8M/137M [00:04<00:10, 10.2Mb/s]
26%|##6      | 35.9M/137M [00:04<00:09, 10.1Mb/s]
27%|##7      | 37.0M/137M [00:05<00:09, 10.5Mb/s]
28%|##7      | 38.1M/137M [00:05<00:09, 10.5Mb/s]
29%|##8      | 39.2M/137M [00:05<00:09, 10.8Mb/s]
29%|##9      | 40.4M/137M [00:05<00:08, 11.0Mb/s]
30%|###      | 41.5M/137M [00:05<00:13, 7.19Mb/s]
31%|###1     | 42.6M/137M [00:05<00:11, 8.07Mb/s]
32%|###1     | 43.7M/137M [00:05<00:10, 8.79Mb/s]
33%|###2     | 44.8M/137M [00:05<00:09, 9.33Mb/s]
34%|###3     | 45.9M/137M [00:06<00:09, 9.29Mb/s]
34%|###4     | 47.0M/137M [00:06<00:09, 9.80Mb/s]
35%|###5     | 48.0M/137M [00:06<00:09, 9.64Mb/s]
36%|###5     | 49.0M/137M [00:06<00:09, 9.64Mb/s]
37%|###6     | 50.0M/137M [00:06<00:09, 9.53Mb/s]
37%|###7     | 51.0M/137M [00:06<00:09, 9.46Mb/s]
38%|###8     | 52.1M/137M [00:06<00:08, 9.76Mb/s]
39%|###8     | 53.2M/137M [00:06<00:08, 10.1Mb/s]
40%|###9     | 54.2M/137M [00:06<00:08, 10.1Mb/s]
40%|####     | 55.2M/137M [00:06<00:08, 10.2Mb/s]
41%|####1    | 56.3M/137M [00:07<00:07, 10.2Mb/s]
42%|####1    | 57.4M/137M [00:07<00:07, 10.5Mb/s]
43%|####2    | 58.5M/137M [00:07<00:07, 10.5Mb/s]
43%|####3    | 59.5M/137M [00:07<00:07, 10.5Mb/s]
44%|####4    | 60.6M/137M [00:07<00:07, 10.5Mb/s]
45%|####5    | 61.7M/137M [00:07<00:07, 10.6Mb/s]
```

46%	#####5	62.8M/137M	[00:07<00:06,	10.7Mb/s]
47%	#####6	63.9M/137M	[00:07<00:06,	10.8Mb/s]
47%	#####7	65.0M/137M	[00:07<00:06,	10.9Mb/s]
48%	#####8	66.1M/137M	[00:07<00:06,	10.9Mb/s]
49%	#####9	67.3M/137M	[00:08<00:06,	11.0Mb/s]
50%	#####9	68.4M/137M	[00:08<00:06,	11.0Mb/s]
51%	#####	69.5M/137M	[00:08<00:06,	11.1Mb/s]
52%	#####1	70.6M/137M	[00:08<00:06,	10.7Mb/s]
52%	#####2	71.7M/137M	[00:08<00:06,	10.4Mb/s]
53%	#####3	72.7M/137M	[00:08<00:06,	9.41Mb/s]
54%	#####3	73.8M/137M	[00:08<00:06,	9.65Mb/s]
55%	#####4	74.8M/137M	[00:08<00:06,	9.82Mb/s]
55%	#####5	75.8M/137M	[00:08<00:06,	9.86Mb/s]
56%	#####6	77.0M/137M	[00:09<00:05,	10.3Mb/s]
57%	#####7	78.1M/137M	[00:09<00:05,	10.7Mb/s]
58%	#####7	79.2M/137M	[00:09<00:05,	10.5Mb/s]
59%	#####8	80.3M/137M	[00:09<00:05,	10.1Mb/s]
59%	#####9	81.3M/137M	[00:09<00:05,	10.2Mb/s]
60%	#####	82.4M/137M	[00:09<00:05,	10.2Mb/s]
61%	#####	83.4M/137M	[00:09<00:05,	10.2Mb/s]
62%	#####1	84.4M/137M	[00:09<00:05,	10.2Mb/s]
62%	#####2	85.5M/137M	[00:09<00:04,	10.5Mb/s]
63%	#####3	86.6M/137M	[00:09<00:04,	10.4Mb/s]
64%	#####4	87.7M/137M	[00:10<00:04,	10.0Mb/s]
65%	#####4	88.8M/137M	[00:10<00:05,	9.55Mb/s]
66%	#####5	89.7M/137M	[00:10<00:04,	9.55Mb/s]
66%	#####6	90.7M/137M	[00:10<00:04,	9.67Mb/s]
67%	#####6	91.7M/137M	[00:10<00:04,	9.76Mb/s]
68%	#####7	92.8M/137M	[00:10<00:04,	10.0Mb/s]
69%	#####8	93.8M/137M	[00:10<00:04,	9.67Mb/s]
69%	#####9	94.9M/137M	[00:10<00:04,	10.0Mb/s]
70%	#####	96.0M/137M	[00:10<00:04,	10.2Mb/s]
71%	#####	97.0M/137M	[00:11<00:04,	9.91Mb/s]
72%	#####1	98.0M/137M	[00:11<00:03,	10.0Mb/s]
72%	#####2	99.2M/137M	[00:11<00:03,	10.3Mb/s]
73%	#####3	100M/137M	[00:11<00:03,	10.1Mb/s]
74%	#####3	101M/137M	[00:11<00:03,	9.23Mb/s]
75%	#####4	102M/137M	[00:11<00:03,	9.58Mb/s]
75%	#####5	103M/137M	[00:11<00:03,	9.76Mb/s]
76%	#####6	104M/137M	[00:11<00:04,	6.73Mb/s]
77%	#####6	105M/137M	[00:12<00:04,	7.37Mb/s]
78%	#####7	106M/137M	[00:12<00:03,	8.17Mb/s]
79%	#####8	108M/137M	[00:12<00:03,	8.99Mb/s]
79%	#####9	109M/137M	[00:12<00:03,	9.38Mb/s]
80%	#####	110M/137M	[00:12<00:02,	9.93Mb/s]
81%	#####	111M/137M	[00:12<00:02,	10.4Mb/s]
82%	#####1	112M/137M	[00:12<00:02,	10.3Mb/s]
83%	#####2	113M/137M	[00:12<00:02,	9.97Mb/s]
83%	#####3	114M/137M	[00:12<00:02,	9.97Mb/s]
84%	#####4	115M/137M	[00:12<00:02,	10.2Mb/s]
85%	#####4	116M/137M	[00:13<00:02,	10.2Mb/s]
86%	#####5	117M/137M	[00:13<00:01,	10.5Mb/s]
86%	#####6	118M/137M	[00:13<00:01,	10.4Mb/s]
87%	#####7	119M/137M	[00:13<00:01,	10.0Mb/s]
88%	#####7	120M/137M	[00:13<00:01,	9.94Mb/s]
89%	#####8	122M/137M	[00:13<00:01,	10.3Mb/s]
90%	#####9	123M/137M	[00:13<00:01,	9.96Mb/s]
90%	#####	124M/137M	[00:13<00:01,	10.1Mb/s]
91%	#####1	125M/137M	[00:13<00:01,	10.3Mb/s]
92%	#####1	126M/137M	[00:14<00:01,	10.5Mb/s]
93%	#####2	127M/137M	[00:14<00:00,	10.6Mb/s]
94%	#####3	128M/137M	[00:14<00:00,	10.5Mb/s]
94%	#####4	129M/137M	[00:14<00:00,	10.8Mb/s]
95%	#####5	130M/137M	[00:14<00:00,	10.7Mb/s]
96%	#####5	131M/137M	[00:14<00:00,	10.2Mb/s]
97%	#####6	132M/137M	[00:14<00:00,	10.2Mb/s]

```
98%|#####7| 134M/137M [00:14<00:00, 10.4Mb/s]
98%|#####8| 135M/137M [00:14<00:00, 10.2Mb/s]
99%|#####9| 136M/137M [00:14<00:00, 10.3Mb/s]
100%|#####9| 137M/137M [00:15<00:00, 10.5Mb/s]
100%|#####| 137M/137M [00:15<00:00, 9.07Mb/s]
[INFO] Beginning extraction
[INFO] Chromium extracted to: C:\Users\Eduardo\AppData\Local\pyppeteer\pyppeteer\local-chromium\588429
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 2522654 bytes to Análise de Culturas.pdf
```