Pregunta 5

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

In [2]:
```
df = pd.read_csv("wine.csv")
```

In [3]:
```
df
```

Out[3]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	NaN	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

178 rows × 15 columns

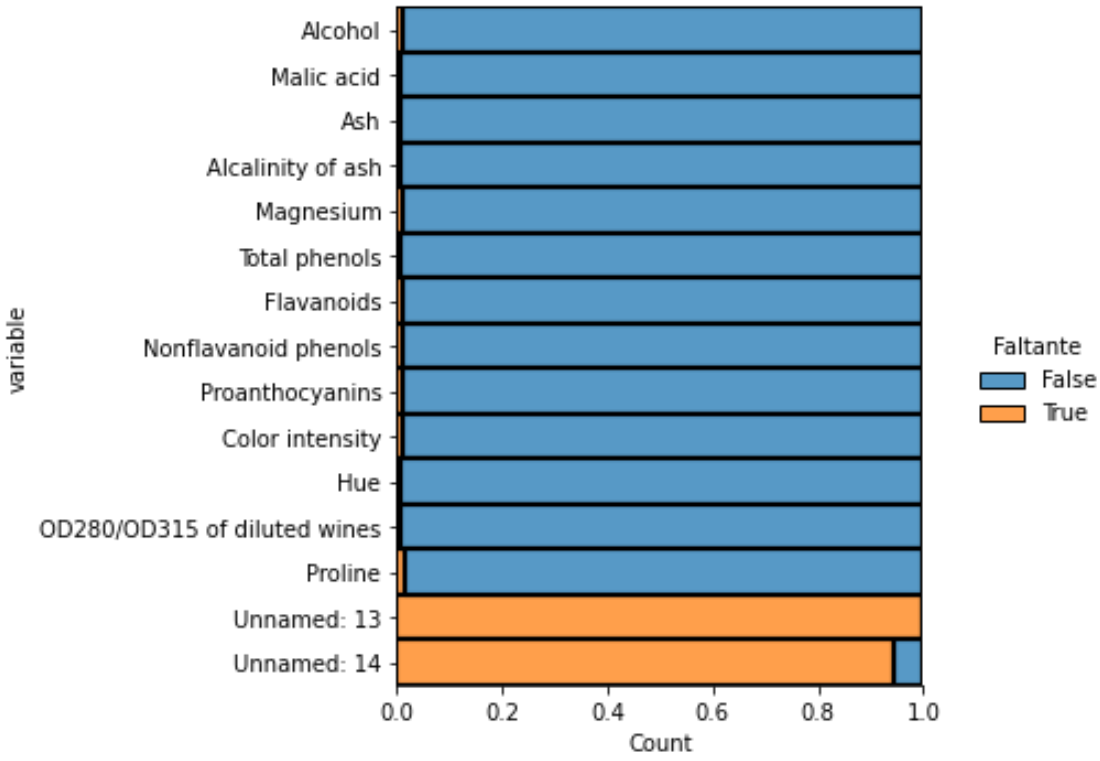
1- (5) Detecta los valores ausentes y reemplázalos con un valor numérico adecuado.

In [4]:
```
plt.figure(figsize=(10,6))
sns.displot(
    data=df.isna().melt(value_name="Faltante"),
    y="variable",
    hue="Faltante",
```

```
    multiple="fill",
    aspect=1.25
)
```

Out[4]: <seaborn.axisgrid.FacetGrid at 0x7ff0e9340cd0>

<Figure size 720x432 with 0 Axes>



In [5]:
```
# Contamos los valores nulos de cada columna
df.isna().sum()
```

Out[5]:
```
Alcohol                          2
Malic acid                       1
Ash                              1
Alcalinity of ash                1
Magnesium                        2
Total phenols                    1
Flavanoids                       2
Nonflavanoid phenols             2
Proanthocyanins                  2
Color intensity                  2
Hue                              1
OD280/OD315 of diluted wines     1
Proline                          3
Unnamed: 13                    178
Unnamed: 14                    168
dtype: int64
```

In [6]:
```
# Eliminamos practicamente las filas que estan con valores nul
del df['Unnamed: 13']
```

```
del df['Unnamed: 14']
```

In [7]:
```
# Remplazamos todos los valores nulos por 0 ya que son pocos l
# a poder realizar mas evaluaciones numericas en el futuro, ad
# el dataframe es de tipo entero.
df = df.fillna(0)
```

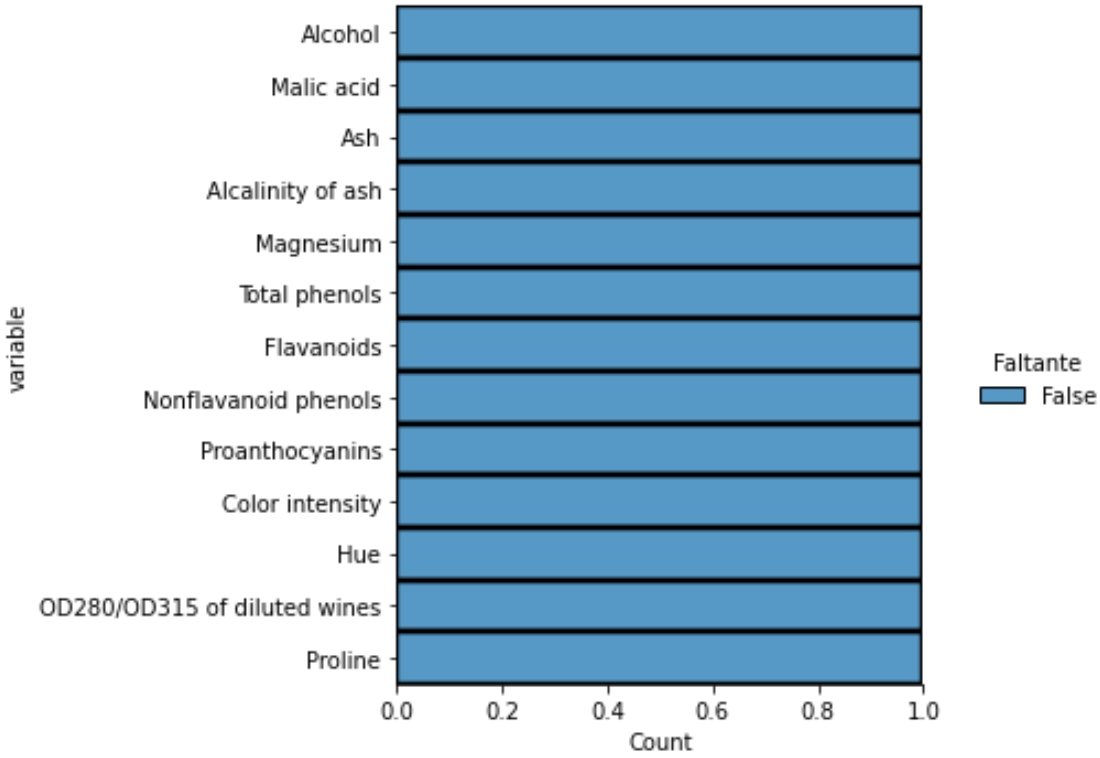
In [8]:
```
#comprobamos que lo que hicimos se realizo.
df.isna().sum()
```

Out[8]:
```
Alcohol                         0
Malic acid                      0
Ash                             0
Alcalinity of ash               0
Magnesium                       0
Total phenols                   0
Flavanoids                      0
Nonflavanoid phenols            0
Proanthocyanins                 0
Color intensity                 0
Hue                             0
OD280/OD315 of diluted wines    0
Proline                         0
dtype: int64
```

In [9]:
```
plt.figure(figsize=(10,6))
sns.displot(
    data=df.isna().melt(value_name="Faltante"),
    y="variable",
    hue="Faltante",
    multiple="fill",
    aspect=1.25
)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x7ff0ec28a8e0>

<Figure size 720x432 with 0 Axes>
```



2- (5) Detecta los valores atípicos y elimínalos.

In [10]:
```
dfanterior = df
len(df.index)
```

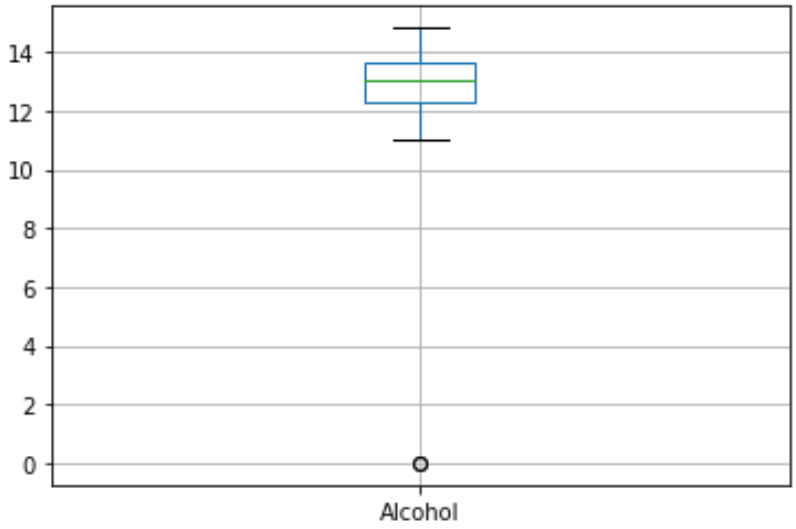
Out[10]:
```
178
```

In [11]:
```
# Se utiliza la distribución normal para buscar los valores at
# y ponemos el numero 3 como umbral, al principio teniamos 178
# y ahora contamos con solo 163. Podriamos hacer la comprobaci
# visualizando un box plot.
from scipy import stats
df = df[(np.abs(stats.zscore(df)) < 3).all(axis=1)]
```

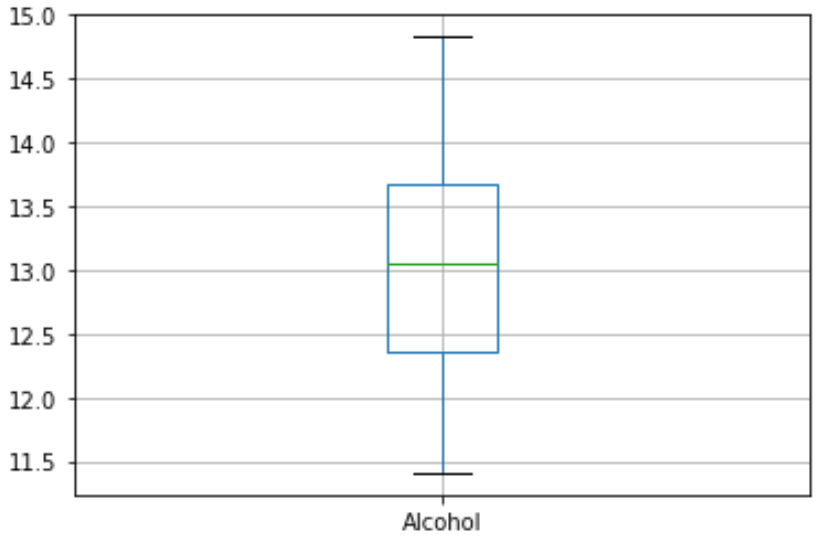
In [12]:
```
len(df.index)
```

Out[12]: 163

In [13]:
```
# Antes de borrar los valores atipicos
boxplot_alcoholAntes = dfanterior.boxplot(column=['Alcohol'])
```



In [14]:
```
# Despues de eliminar los calores atipicos
boxplot_alcoholAhora = df.boxplot(column=['Alcohol'])
```



3- (5) ¿Es necesario transformar columnas?, en caso que si, transforma las columnas y justifica el método utilizado.

En esta etapa yo no lo recomendaria ya que no conozo los rangos adecuados que tienen que tener los componentes de vino, pero seria necesario hacerlo para una

interpretación mejor de los datos en un futuro y utilizar metodos binning como el binning para reducir la cantidad de datos por categoria

Pregunta 6

1 - (15) Realiza un análisis descriptivo de los datos. Interpeta los resultados.

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 163 entries, 0 to 177
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Alcohol                       163 non-null    float64
 1   Malic acid                    163 non-null    float64
 2   Ash                           163 non-null    float64
 3   Alcalinity of ash             163 non-null    float64
 4   Magnesium                     163 non-null    float64
 5   Total phenols                 163 non-null    float64
 6   Flavanoids                    163 non-null    float64
 7   Nonflavanoid phenols          163 non-null    float64
 8   Proanthocyanins               163 non-null    float64
 9   Color intensity               163 non-null    float64
 10  Hue                           163 non-null    float64
 11  OD280/OD315 of diluted wines  163 non-null    float64
 12  Proline                       163 non-null    float64
dtypes: float64(13)
memory usage: 17.8 KB
```

Si vemos los cambios que se han hecho desde el otro ejercicio podemos observar que la cantidad de datos cambio de 178 a 163 ya que se eliminaron los outliers, también podemos observar que no existen datos nulos porque estos fueron remplazados por 0 y en su mayoria se volvieron outliers entonces fueron eliminados de igual maner, tambien vemos que las 13 columnas son de tipo flotante y esto nos ayuda a ver si podremos manipularlas con diferentes graficos en el futuro.

In [16]:
```
df.describe()
```

Out[16]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Fla
count	163.000000	163.000000	163.000000	163.000000	163.000000	163.000000	163

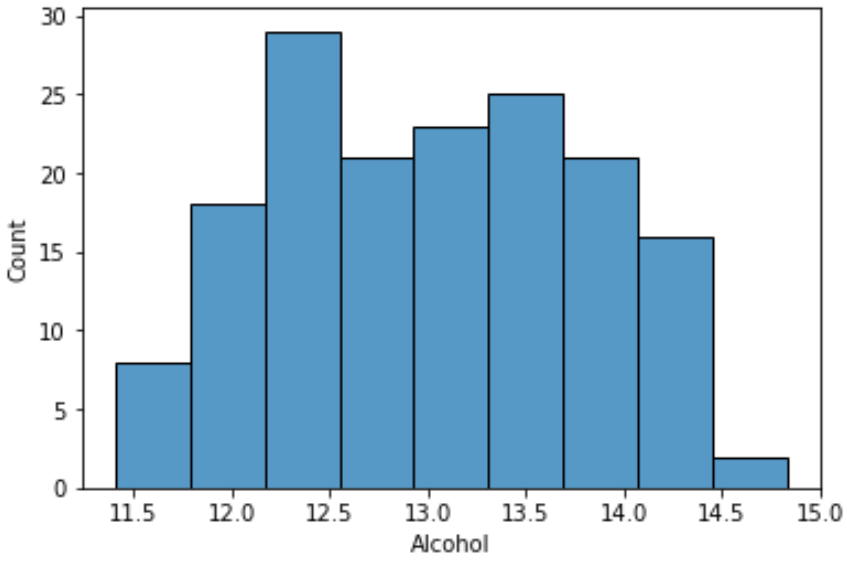
	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Fla
mean	13.016994	2.286626	2.368957	19.444785	99.490798	2.293558	2
std	0.792575	1.102891	0.255972	3.267236	13.789586	0.632396	1
min	11.410000	0.000000	1.700000	11.200000	70.000000	0.980000	0
25%	12.365000	1.580000	2.230000	17.150000	88.000000	1.710000	1
50%	13.050000	1.830000	2.360000	19.100000	98.000000	2.320000	2

Si vemos los percentiles de la mayoria de los datos podemos observar que estan en un rango de bastante analisis ya que no existen datos outliers y al mismo tiempo corresponden con los demas datos de promedio, minimo y maximo.

2 - (12)¿Cómo están distribuidos los datos para Alcohol, Alcalinity, Hue? Utiliza histogramas para mostrarlo y realiza la interpretación de los mismos

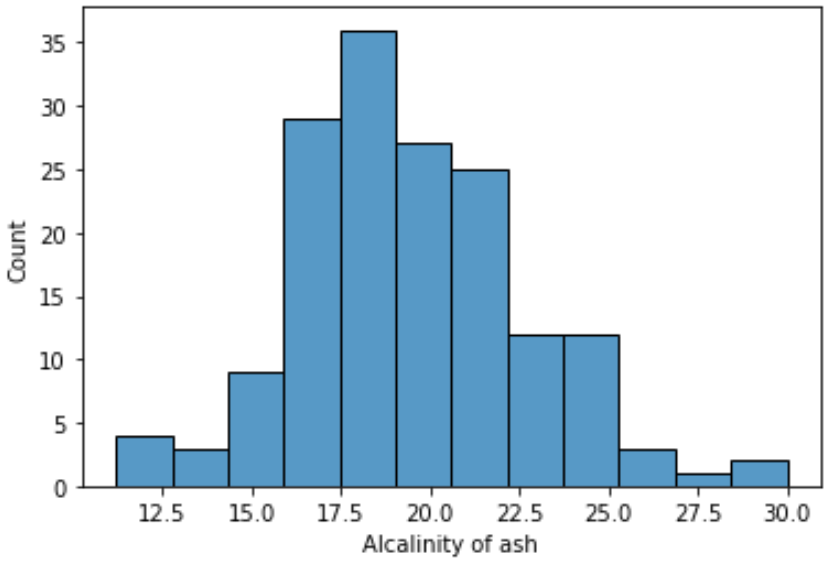
In [17]:
```
sns.histplot(data=df["Alcohol"])
```

Out[17]:
```
<AxesSubplot:xlabel='Alcohol', ylabel='Count'>
```



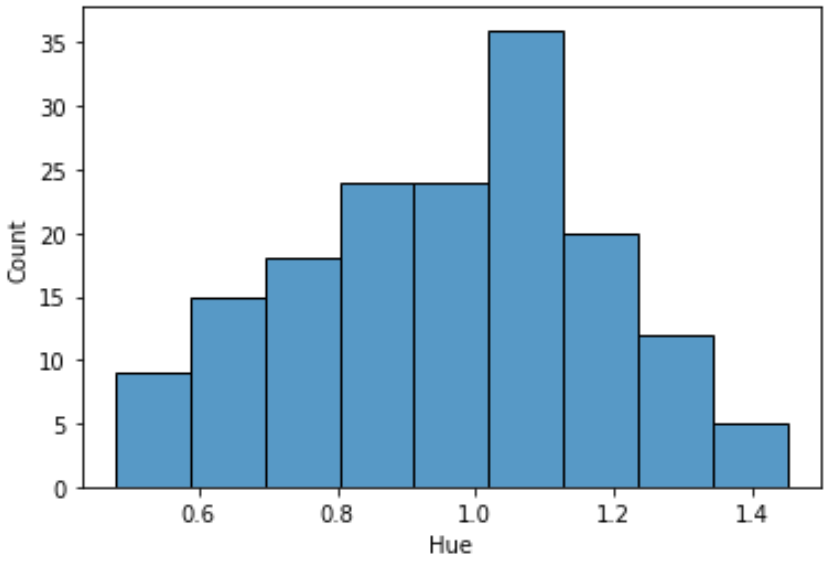
In [18]:
```
sns.histplot(data=df["Alcalinity of ash"])
```

Out[18]:
```
<AxesSubplot:xlabel='Alcalinity of ash', ylabel='Count'>
```



In [19]:
```
sns.histplot(data=df["Hue"])
```

Out[19]:
```
<AxesSubplot:xlabel='Hue', ylabel='Count'>
```

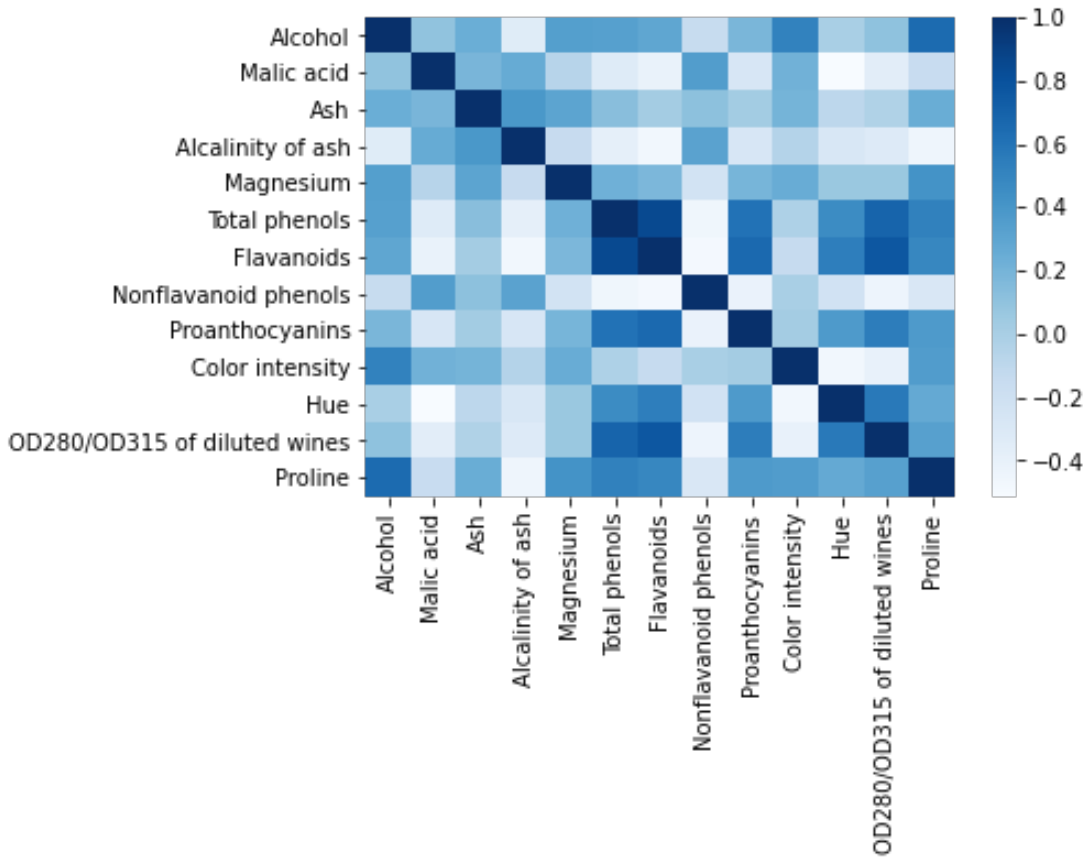


En los diferentes histogramos podemos notar una vez más que los datos estan bien acondicionados, ya que se puede observar que los datos tienden ha formar una campana y en su mayoria podriamos decir que los datos estan casi en la media de los histogramas y no tenemos pocos datos. También aunque no lo he comentado a mi me gustaria normalizar todos los datos a porcentajes ya que con esto podria realizar una interpretación mejor de los datos pero para poder hacer esto necesitaria saber especificamente datos mas especificos del dataframe.

3- (15)Realiza un análisis de correlación de los datos. Realiza una interpretación del mismo.

In [20]:
```
corr = df.corr()
sns.heatmap(corr, cmap="Blues", annot=False)
```

Out[20]: <AxesSubplot:>



Al observar el mapa de calor de correlación podenmos notar a primera vista que la mayoria de las correlaciónes lineales existentes son negativas, entonces esto nos dice que existen varios datos que tienen una fuerte relacion negativa lineal entre sus variables.