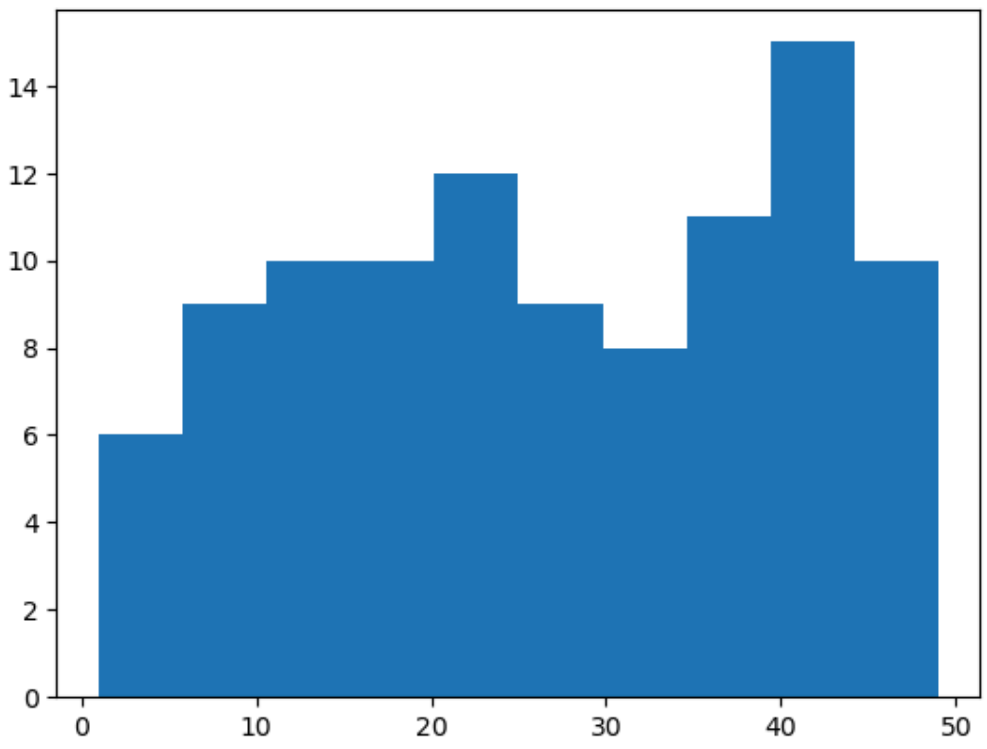
Otro tipos de gráfico

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
data = np.random.randint(1,50,100)
```

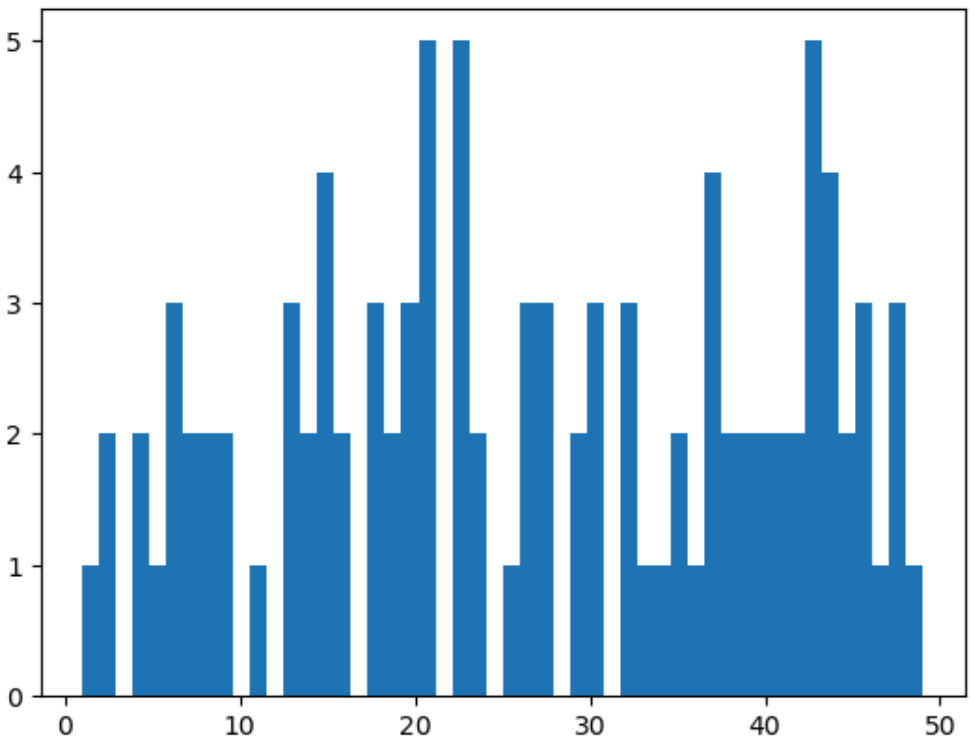
Histograma

In [ ]:
```
#Analizar distribución y frecuencia
#histograma
plt.hist(data)
#bins = 10 por default
plt.show()
```

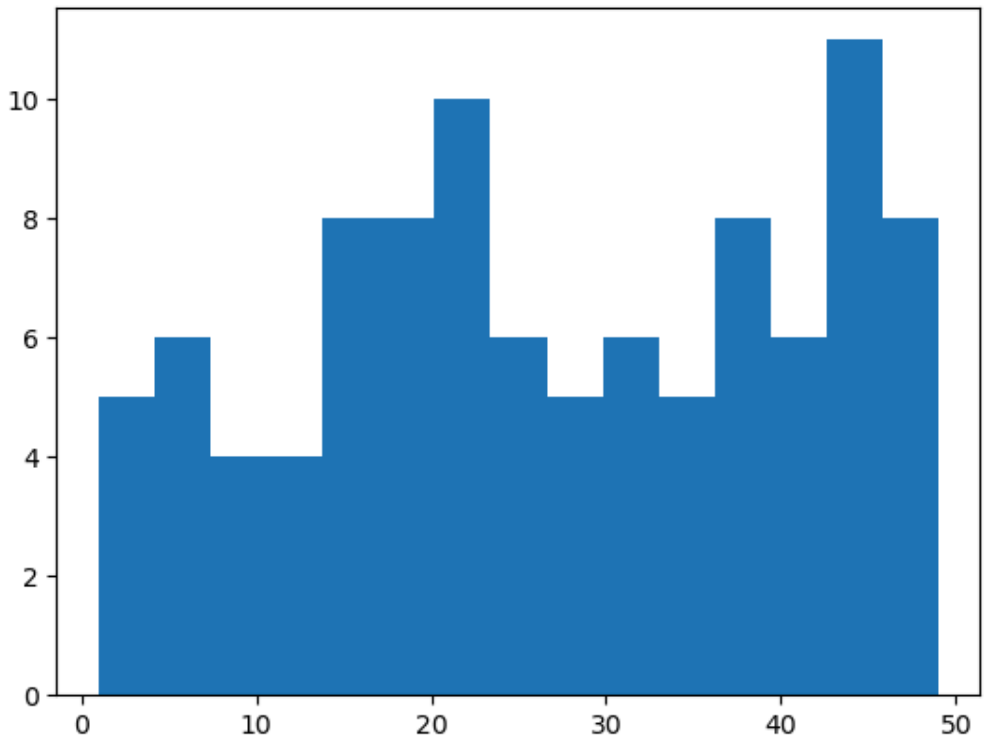


Bins = Número de barras en la cuál va a llegar la distribución de datos

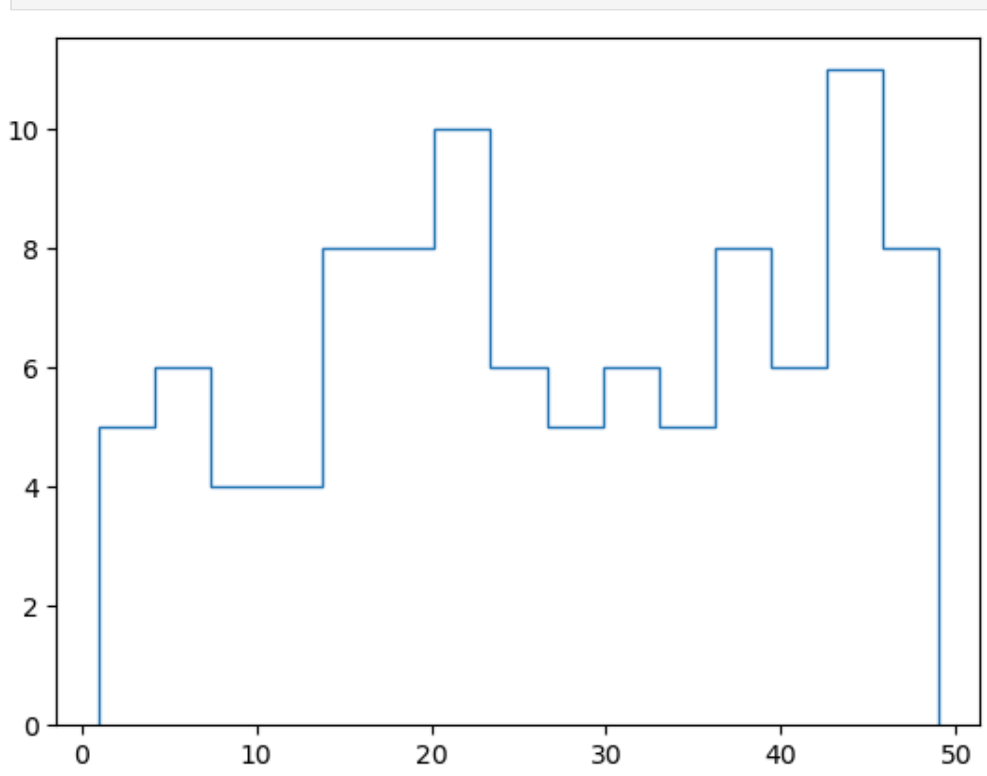
In [ ]:
```
#Jugar cn una variable importante
#Bins en histogramas
plt.hist(data,bins=50)
plt.show()
```



In [ ]:
```
#Modificando parametros
plt.hist(data,bins=15,histtype='bar')
plt.show()
```



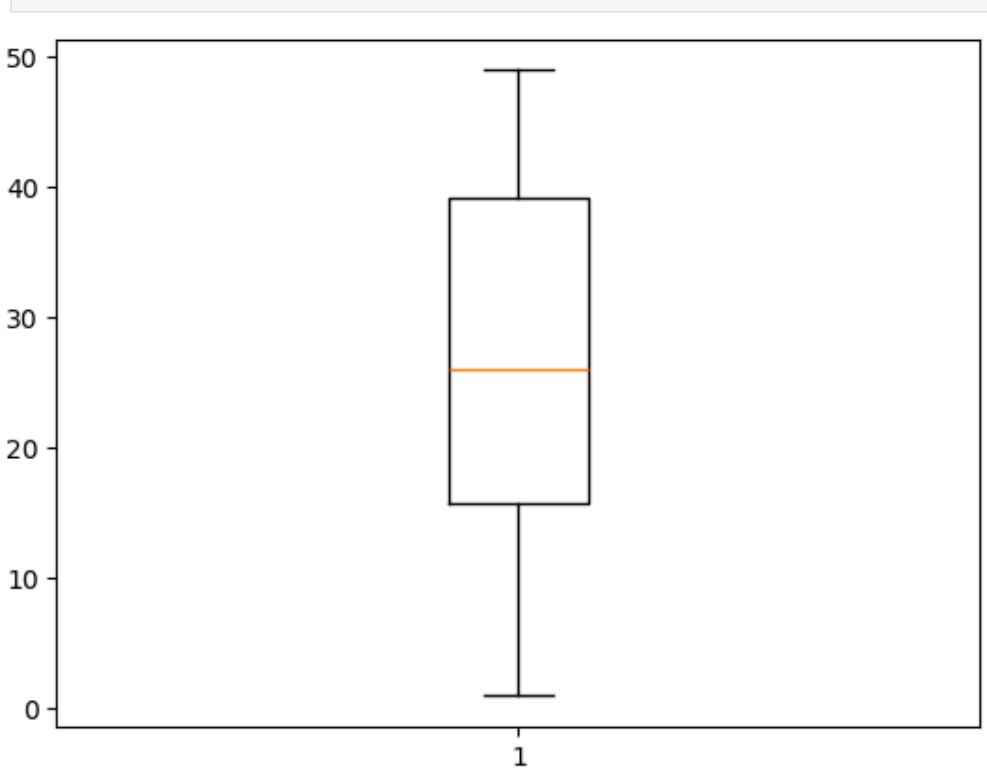
In [ ]:
```
#Modificando parametros
plt.hist(data,bins=15,histtype='step')
plt.show()
```



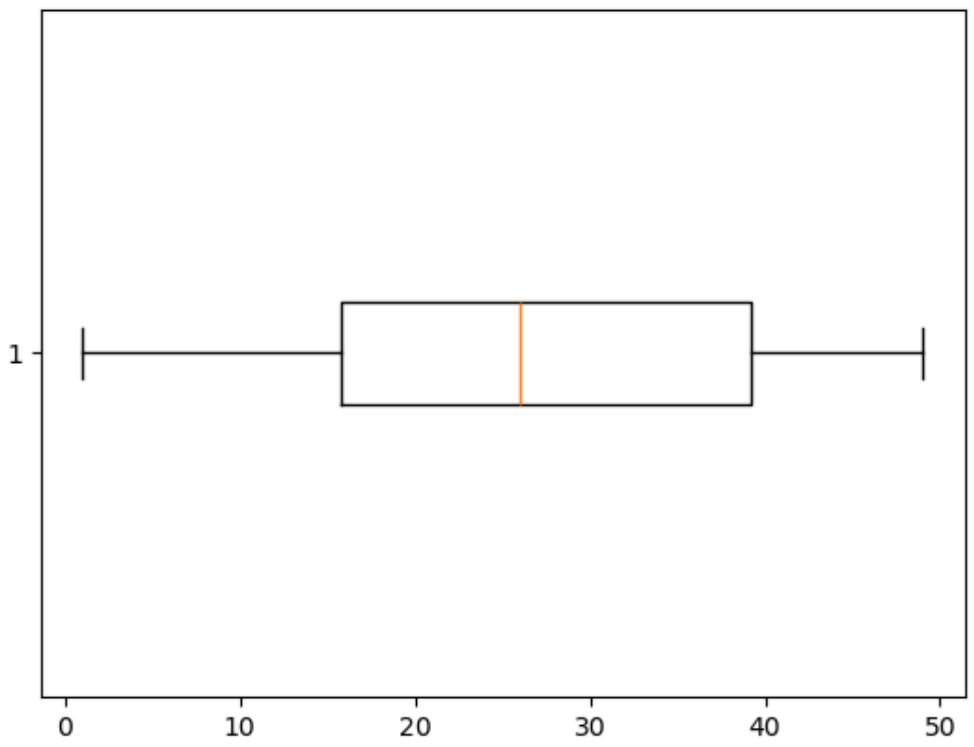
Boxplot

Grafico de cajas y bigotes

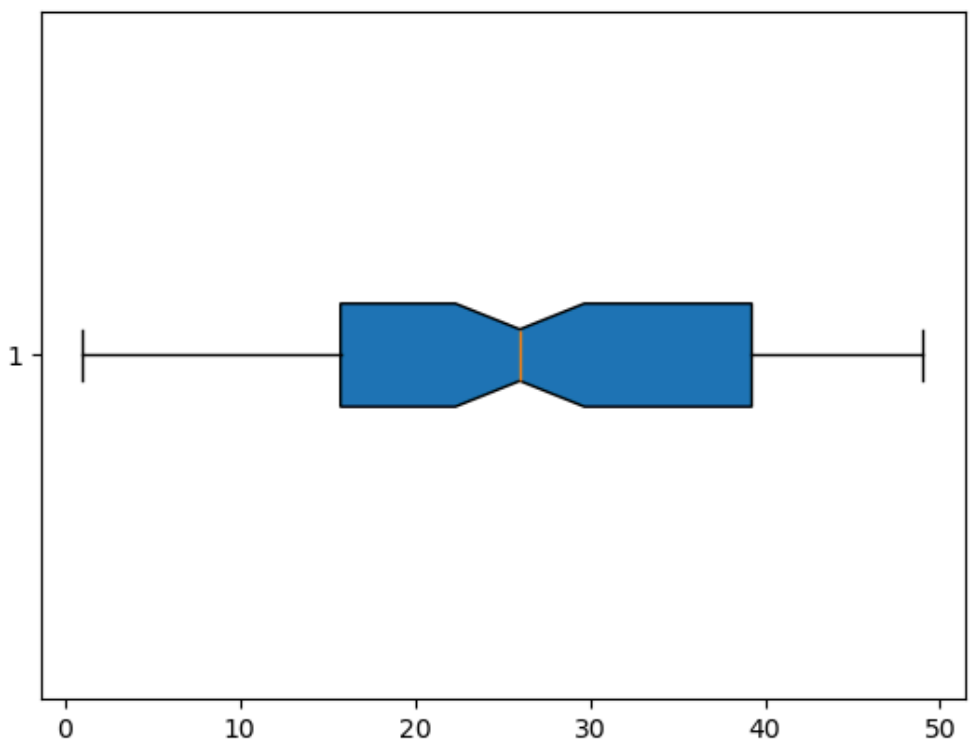
In [ ]:
```
#Boxplot
plt.boxplot(data)
plt.show()
```



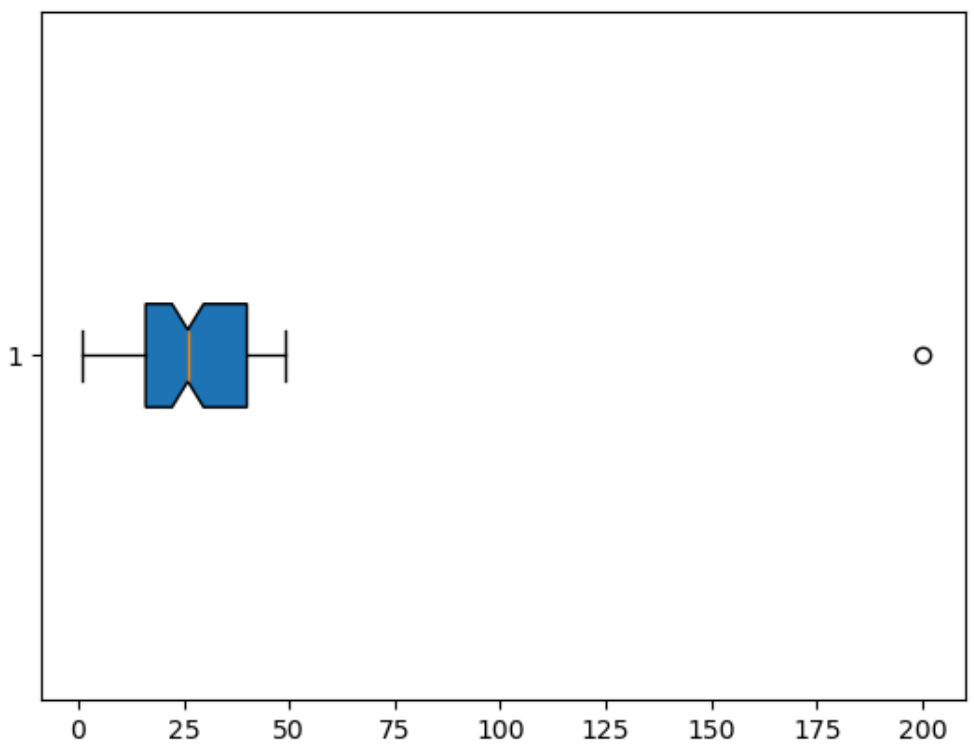
In [ ]:
```
#Explorando propiedades
plt.boxplot(data,vert=False)
plt.show()
```



In [ ]:
```
#Explorando propiedades
plt.boxplot(data,vert=False,patch_artist=True,notch=True)
#Notch me dice como se encuentran los datos en distribuación
plt.show()
```



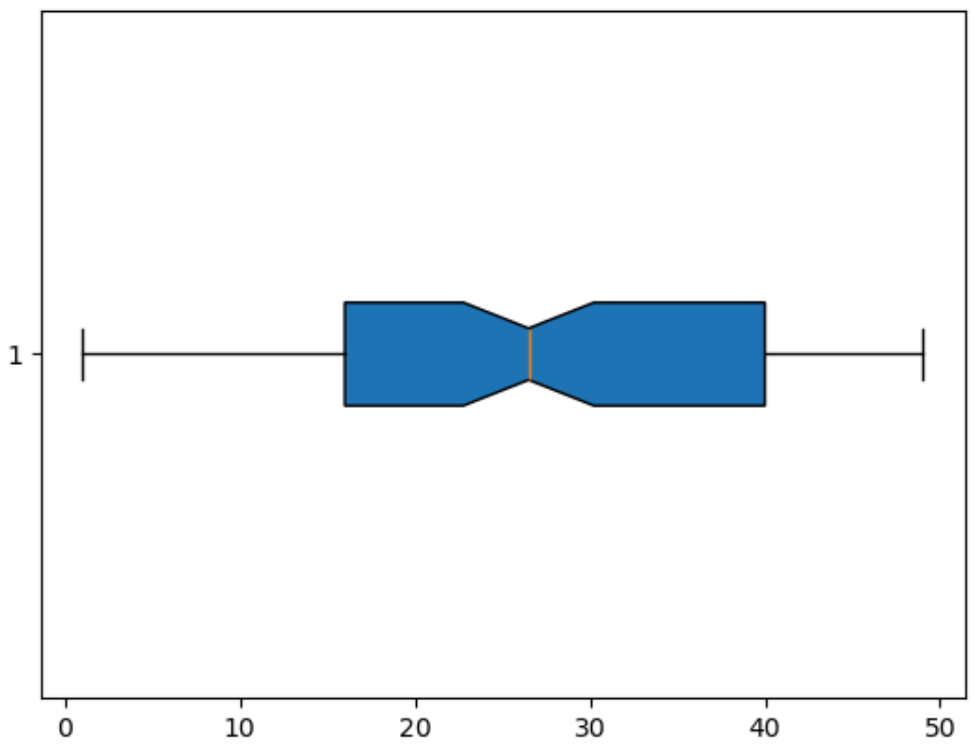
In [ ]:
```
#Agregando data de tipo outlayer (con anomalias)
data=np.append(data,200)
#Explorando propiedades
plt.boxplot(data,vert=False,patch_artist=True,notch=True)
#Notch me dice como se encuentran los datos en distribuación
plt.show()
```



Aquí toma en cuenta el data anomalo. Podemos quitarlo con la sentencia:
`showfliers=False`

In [ ]:
```
#Agregando data de tipo outlayer (con anomalias)
data=np.append(data,200)
#Explorando propiedades
plt.boxplot(data,vert=False,patch_artist=True,notch=True,showfliers=False)
```

```
#Notch me dice como se encuentran los datos en distribuación
plt.show()
```

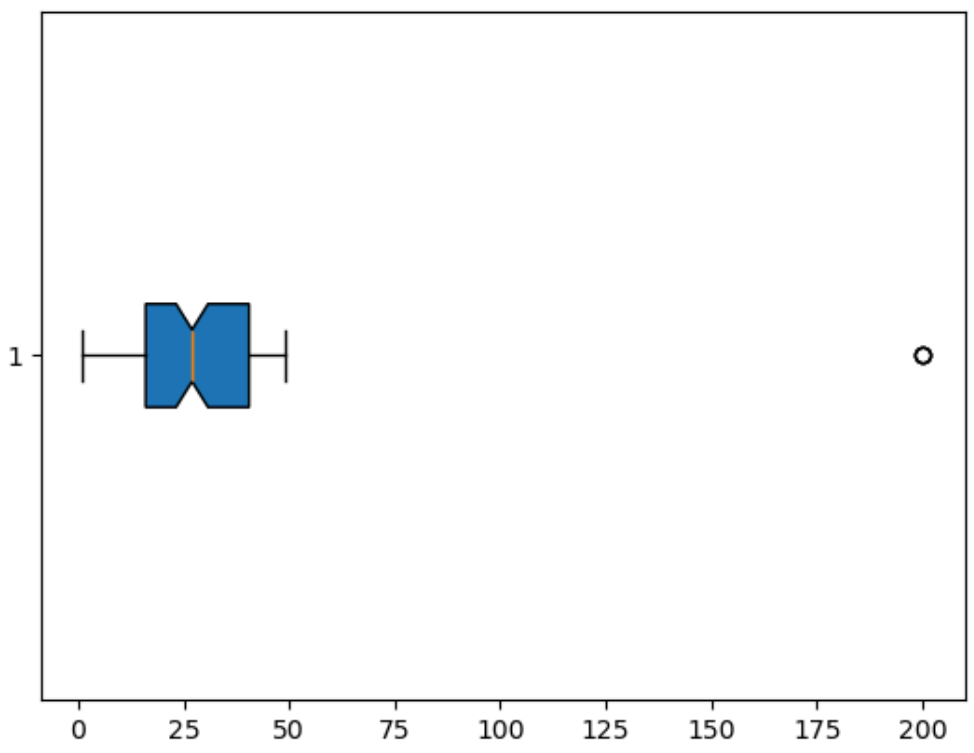


Si le quito el show out layers

Me muestra todos los datos con los que estoy trabajando.

showfliers=True

In [ ]:
```
#Agregando data de tipo outlayer (con anomalias)
data=np.append(data,200)
#Explorando propiedades
plt.boxplot(data,vert=False,patch_artist=True,notch=True,showfliers=True)
#Notch me dice como se encuentran los datos en distribuación
plt.show()
```

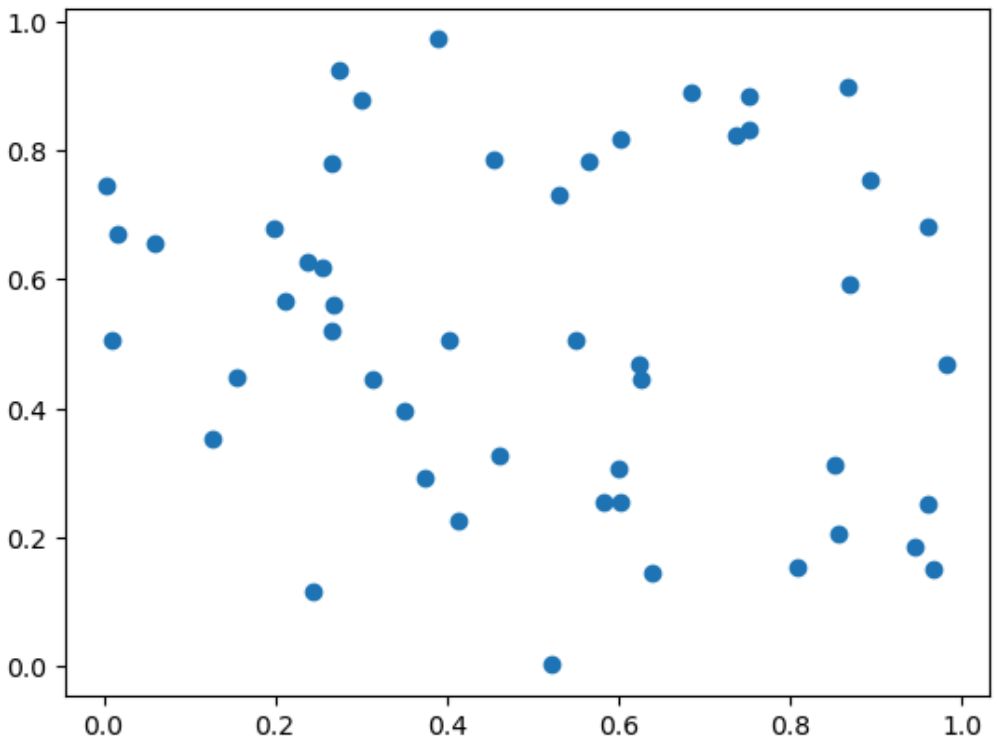


Scatterplot (Dispersión)

In [ ]:
```
N = 50
x = np.random.rand(N)
y = np.random.rand(N)
area = (30*np.random.rand(N))**2
colors = np.random.rand(N)
```

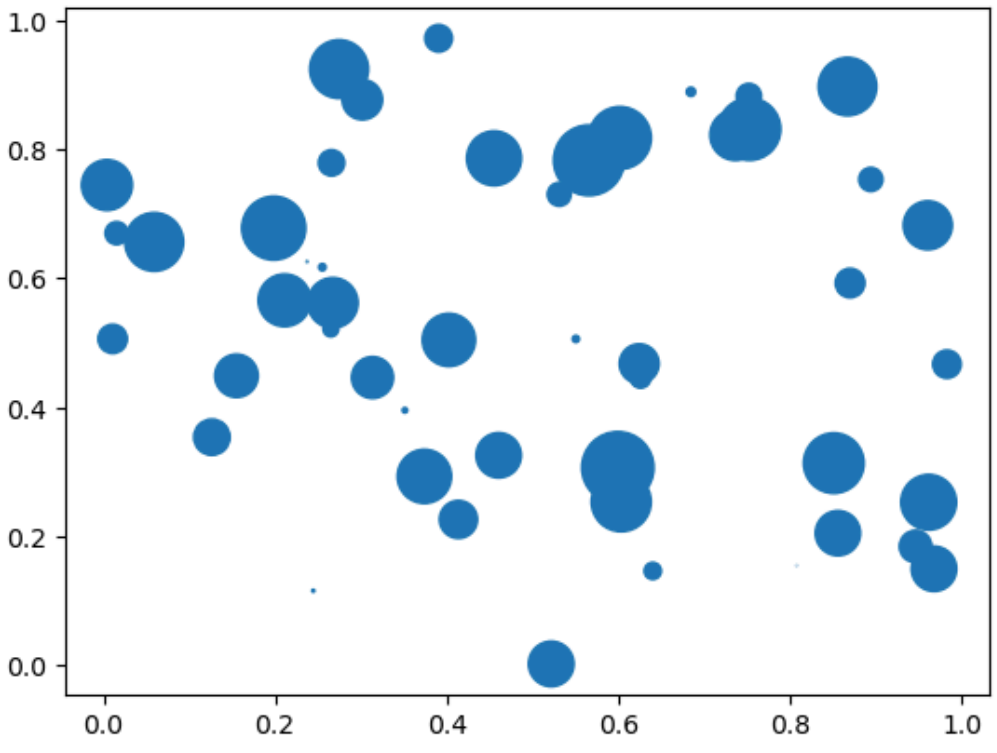
In [ ]:
```
#Graficando Scatter ()
plt.scatter(x,y)
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x7f88e5721ca0>
```



In [ ]:
```
#Graficando Scatter ()
#Agregando más variable
plt.scatter(x,y,s=area)
```

Out[ ]: <matplotlib.collections.PathCollection at 0x7f88e59bad50>



Explicación:

El tamaño de las burbujas va a tener referencia a una 3ra variable que es: area ¿Qué pasa si agrego otra variable?

In [ ]:
```
#Graficando Scatter ()
#Agregando más variable
plt.scatter(x,y,s=area,c=colors)
```

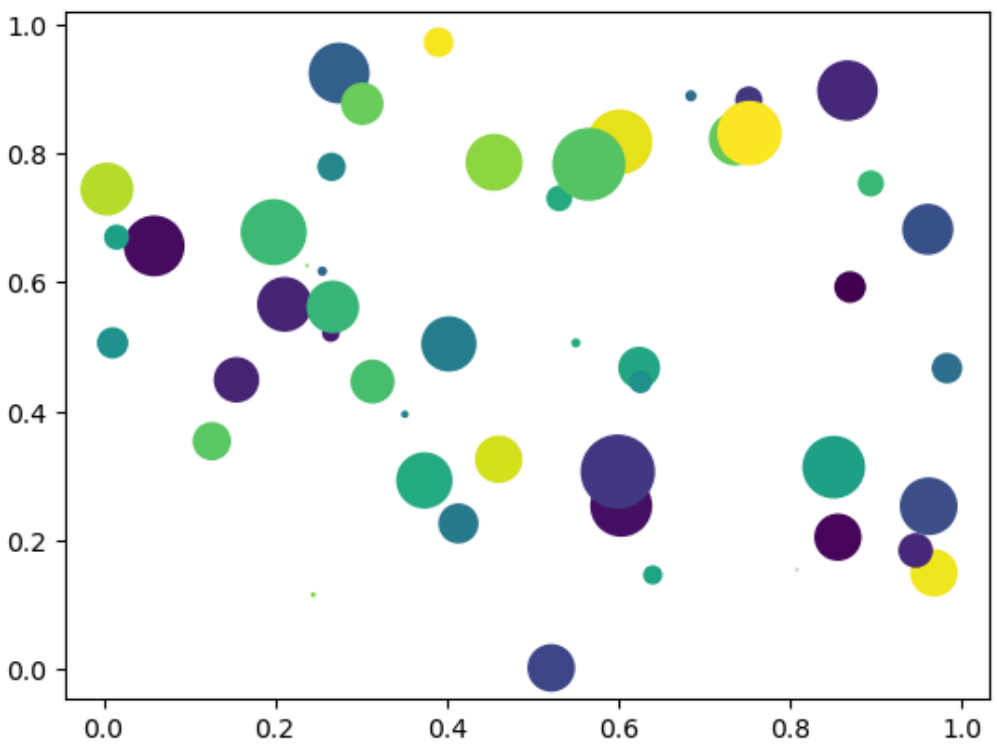
Out[ ]: <matplotlib.collections.PathCollection at 0x7f88e61a7200>

Con esto de los colores me podría servir para hacer una clasificación de lo que estoy haciendo.

In [ ]:
```
#Cambiando markers
#Graficando Scatter ()
```

```
plt.scatter(x,y,s=area,c=colors,marker='o')
```

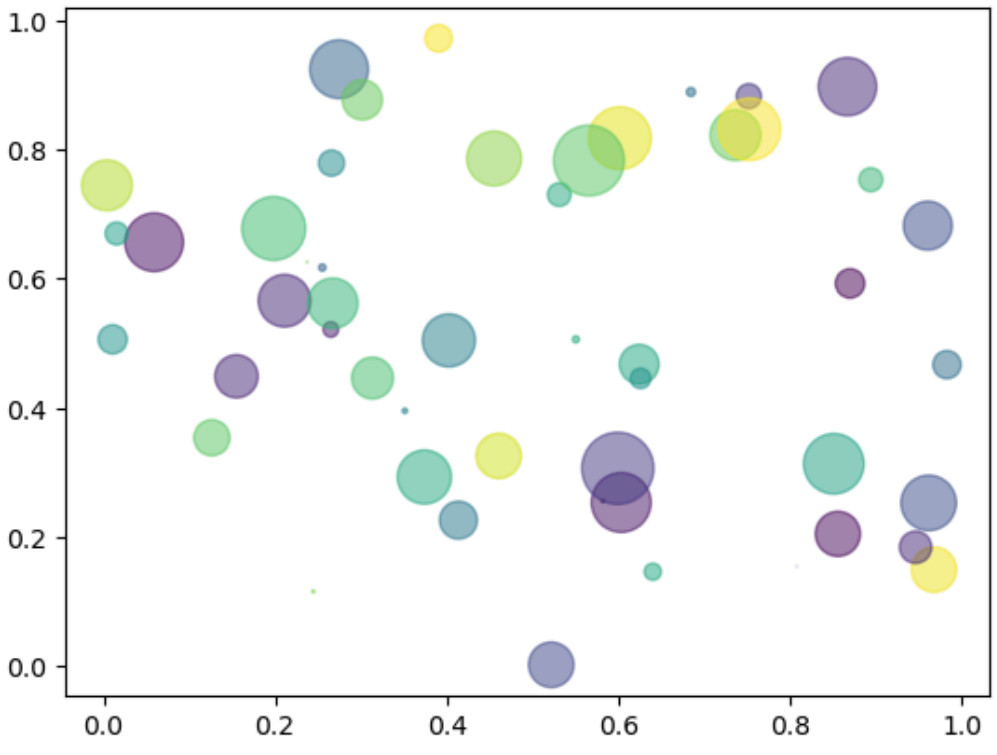
Out[ ]: <matplotlib.collections.PathCollection at 0x7f88e594bec0>



Marker	Tipo
'x'	x
'p'	poligono
's'	Cuadrado
'd'	Diamante
'^'	Triangulo hacia arriba
'v'	Triangulo hacia abajo
'<'	Triangulo hacia la izquierda
'>'	Triangulo hacia la derecha
'o'	Circulo
'|'	Palito
'*'	Estrella
'+'	Cruz
'.'	Puntos pequeños

Ajustando Alpha

In [ ]:
```
#Cambiando markers
#Graficando Scatter ()
plt.scatter(x,y,s=area,c=colors,marker='o',alpha=0.5)
plt.show()
```



De esta manera puedo ver si hay puntos o datos sobrepuestos

Referencias:

- La guía completa sobre gráficos