Distribuciones

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
sns.get_dataset_names()
```

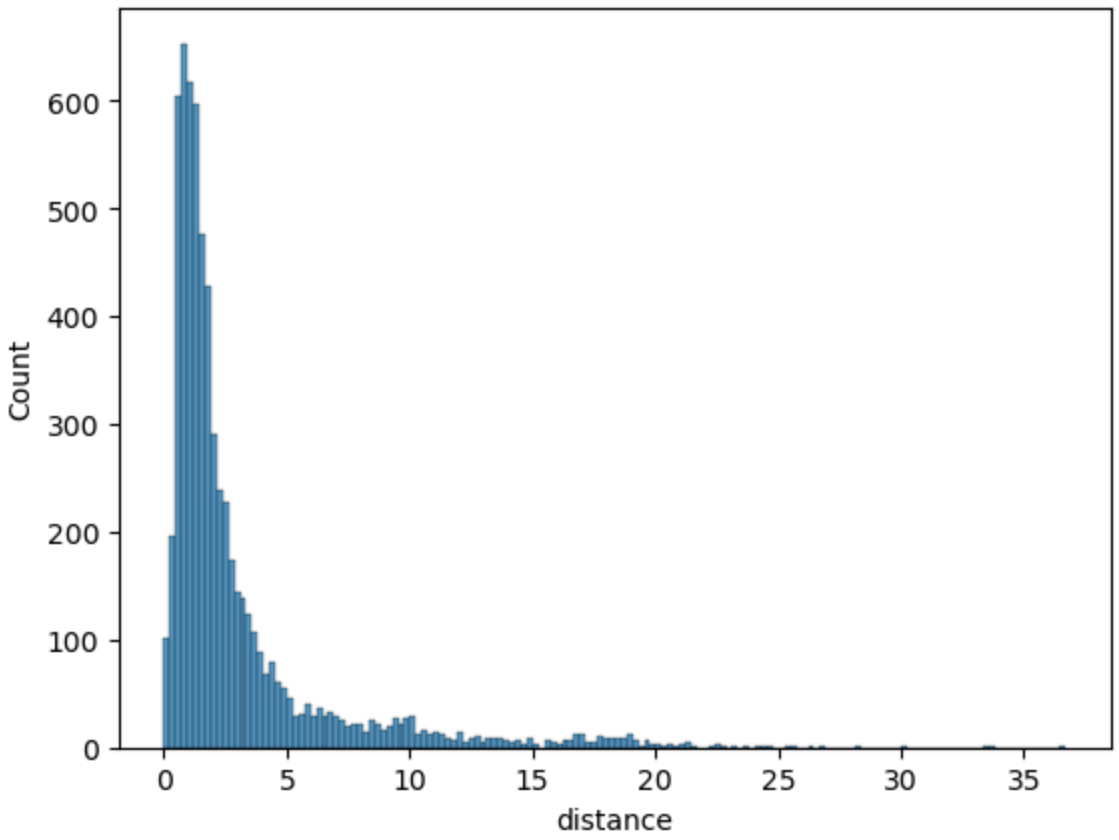
Out[ ]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

In [ ]:
```
#Escogeré el Dataset de taxis
taxis = sns.load_dataset('taxis')
taxis.head(3)
```

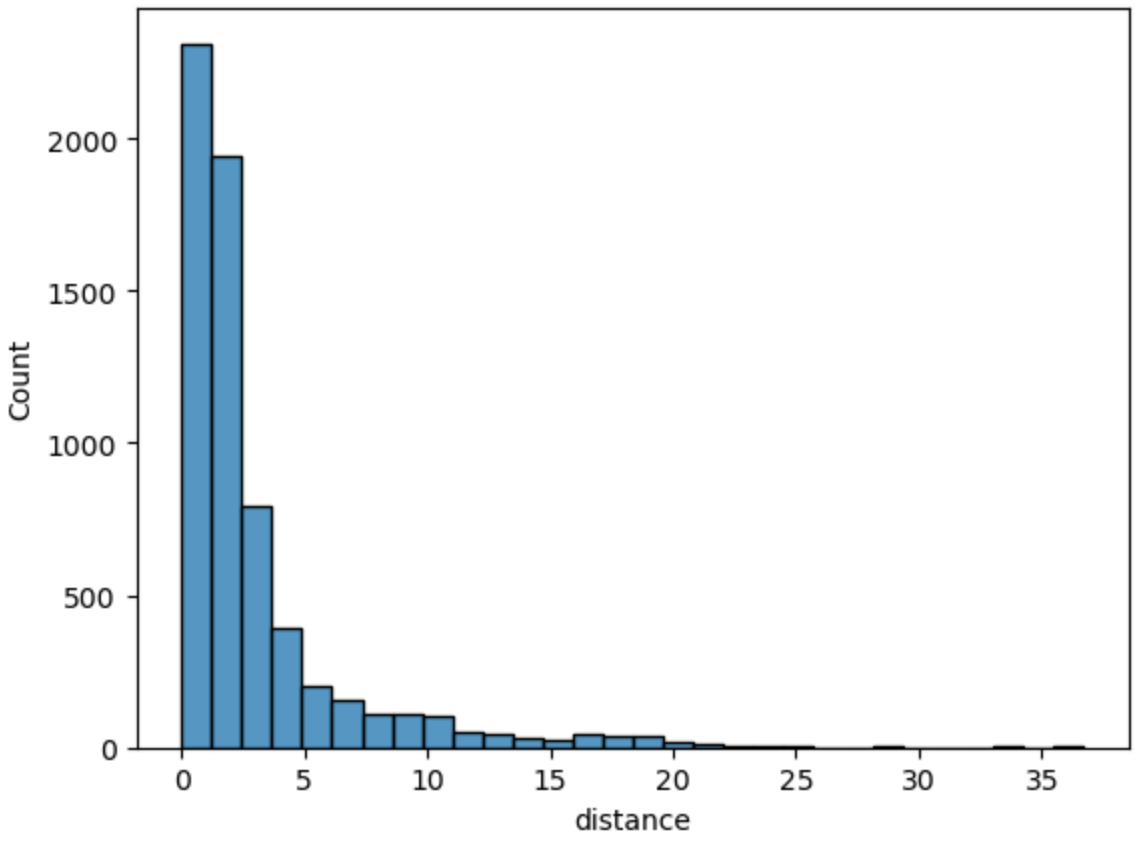
Out[ ]:

	pickup	dropoff	passengers	distance	fare	tip	tolls	total	color	payment	pickup_zone	dropoff_zone	pickup_boroug
0	2019-03-23 20:21:09	2019-03-23 20:27:24	1	1.60	7.0	2.15	0.0	12.95	yellow	credit card	Lenox Hill West	UN/Turtle Bay South	Manhatta
1	2019-03-04 16:11:55	2019-03-04 16:19:00	1	0.79	5.0	0.00	0.0	9.30	yellow	cash	Upper West Side South	Upper West Side South	Manhatta
2	2019-03-27 17:53:01	2019-03-27 18:00:25	1	1.37	7.5	2.36	0.0	14.16	yellow	credit card	Alphabet City	West Village	Manhatta

In [ ]:
```
#Implementando graficos para distribuciones
sns.histplot(data=taxis,x='distance')
plt.show()
```

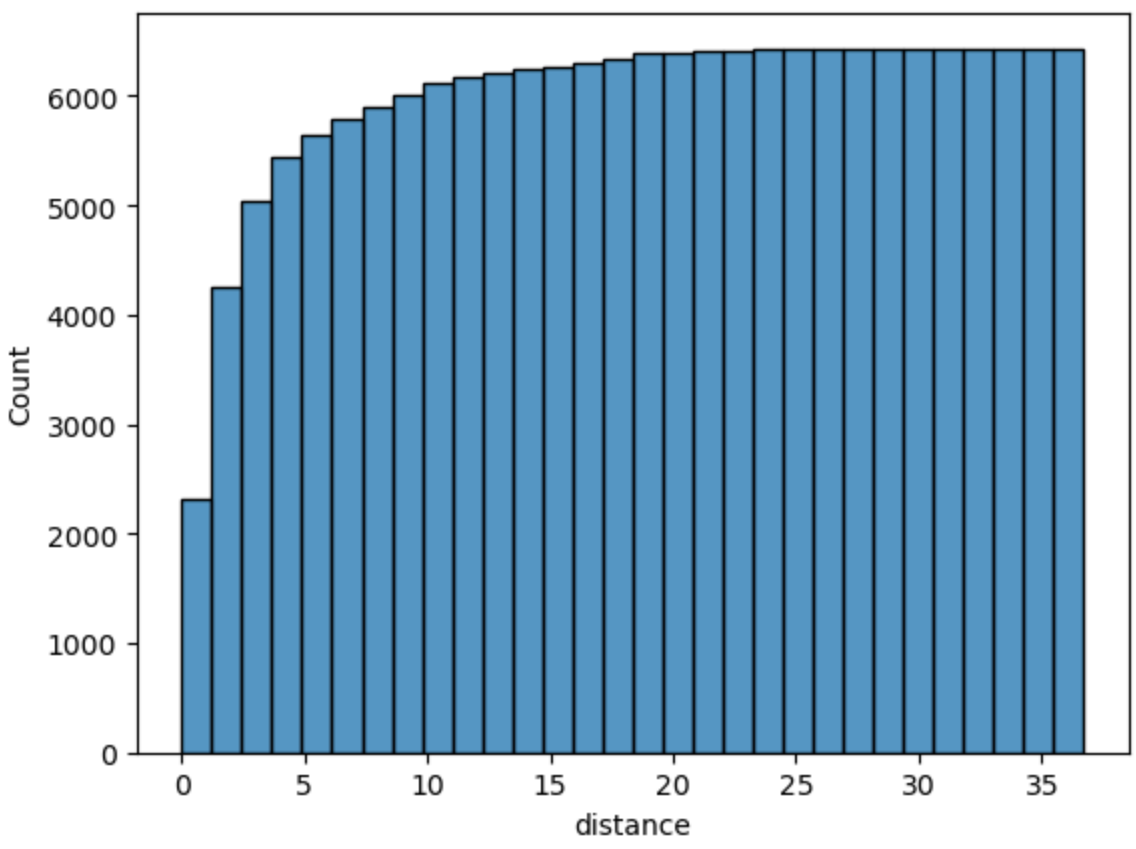


In [ ]:
```
#Agrupando con bins para obtener
#cierto número de barras
sns.histplot(data=taxis,x='distance',bins=30)
plt.show()
```

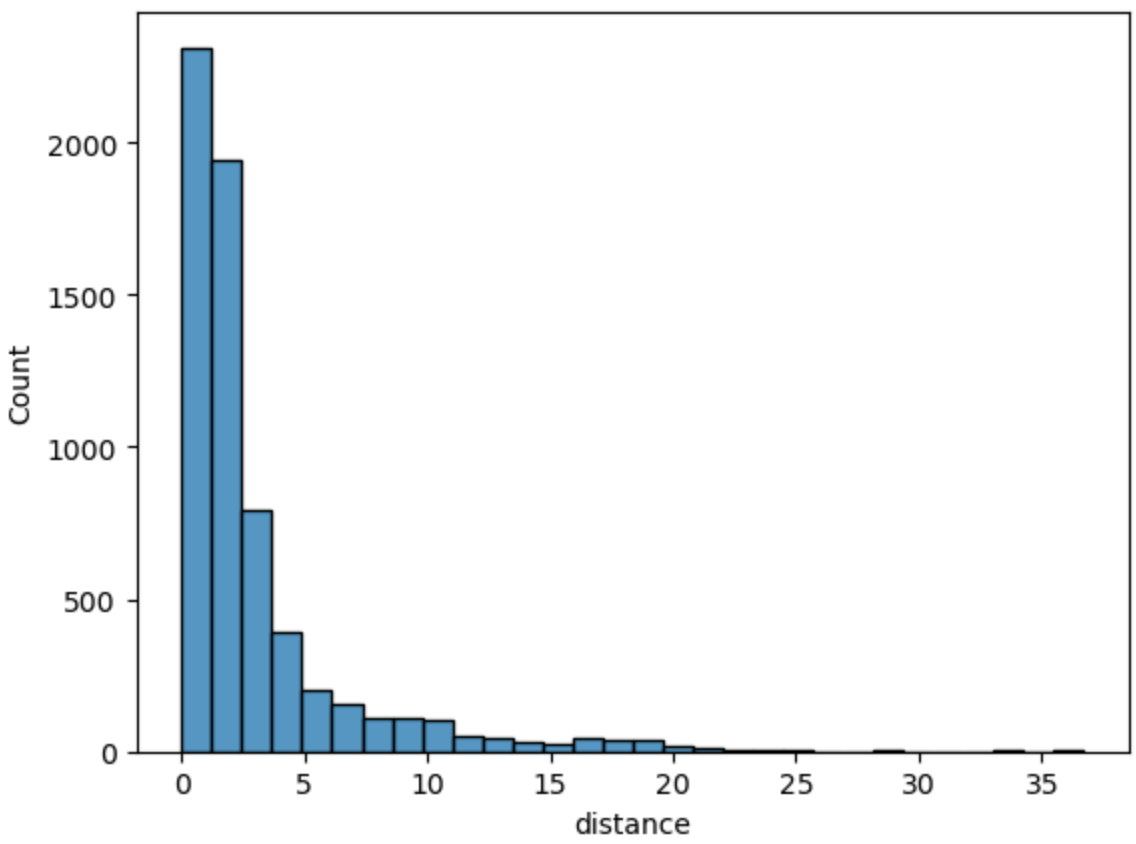


Para ver el comportamiento acumulativo

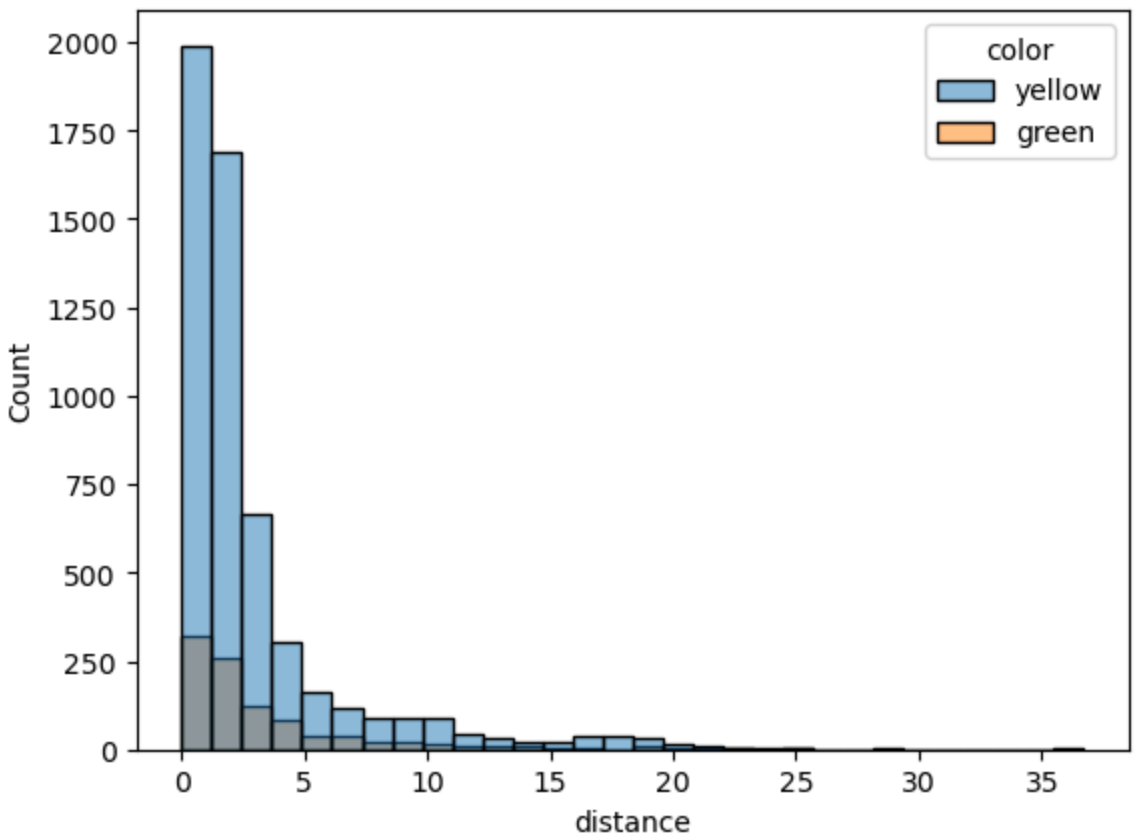
In [ ]:
```
#Seteandolo en verdadero
sns.histplot(data=taxis,x='distance',bins=30,cumulative=True)
plt.show()
```



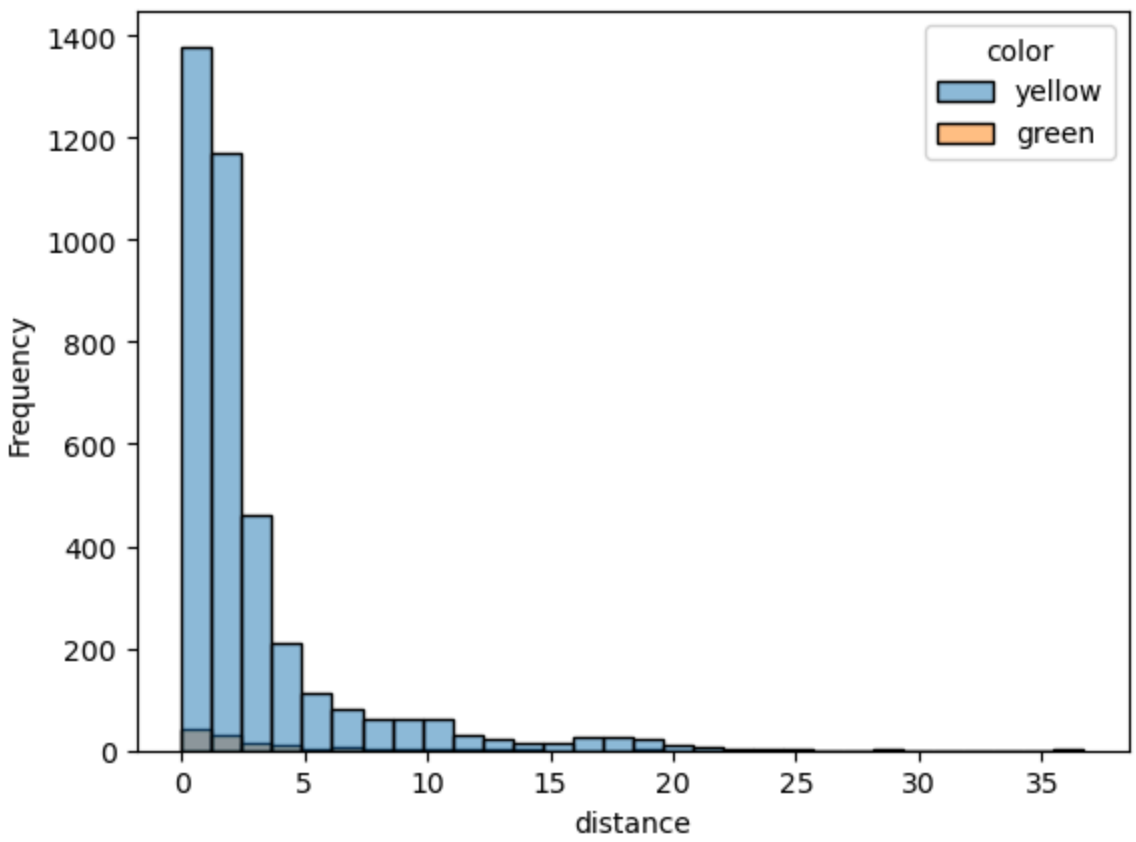
In [ ]:
```
#Seteandolo en Falso
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False)
plt.show()
```



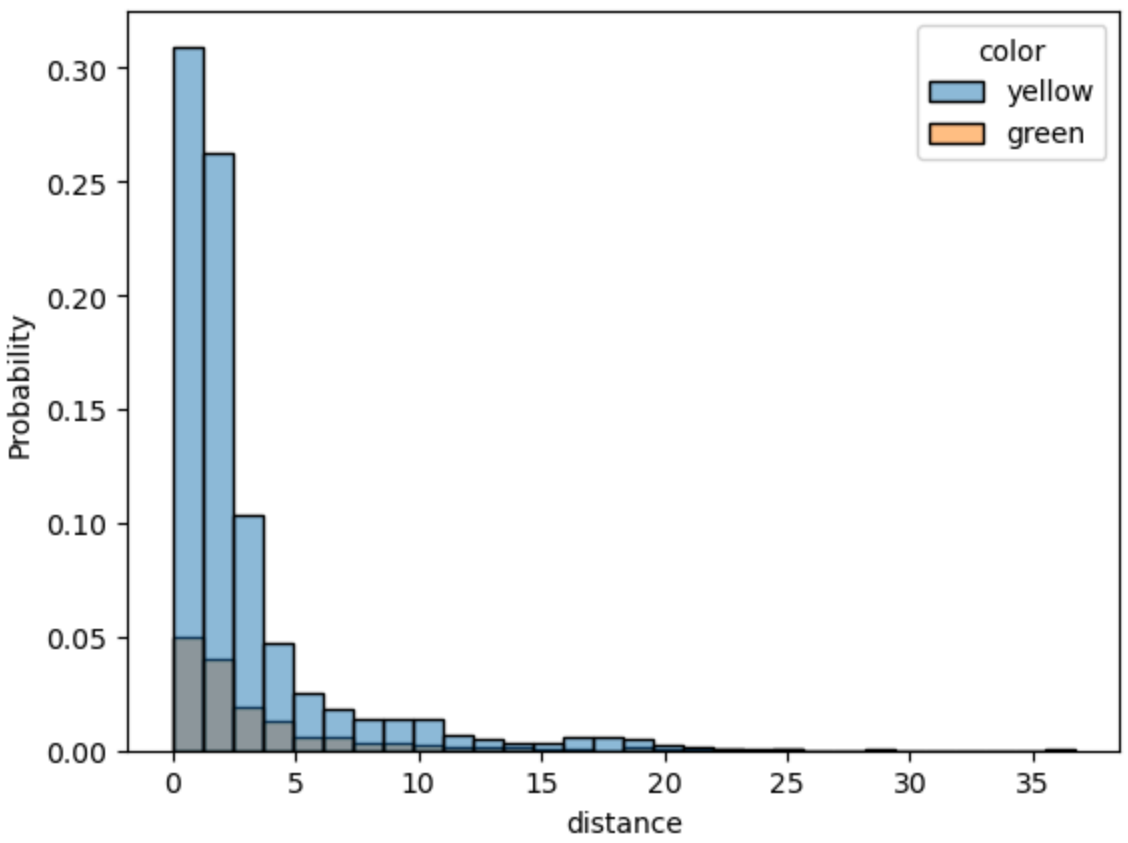
In [ ]:
```
#Aplicando un hue por color
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color')
plt.show()
```



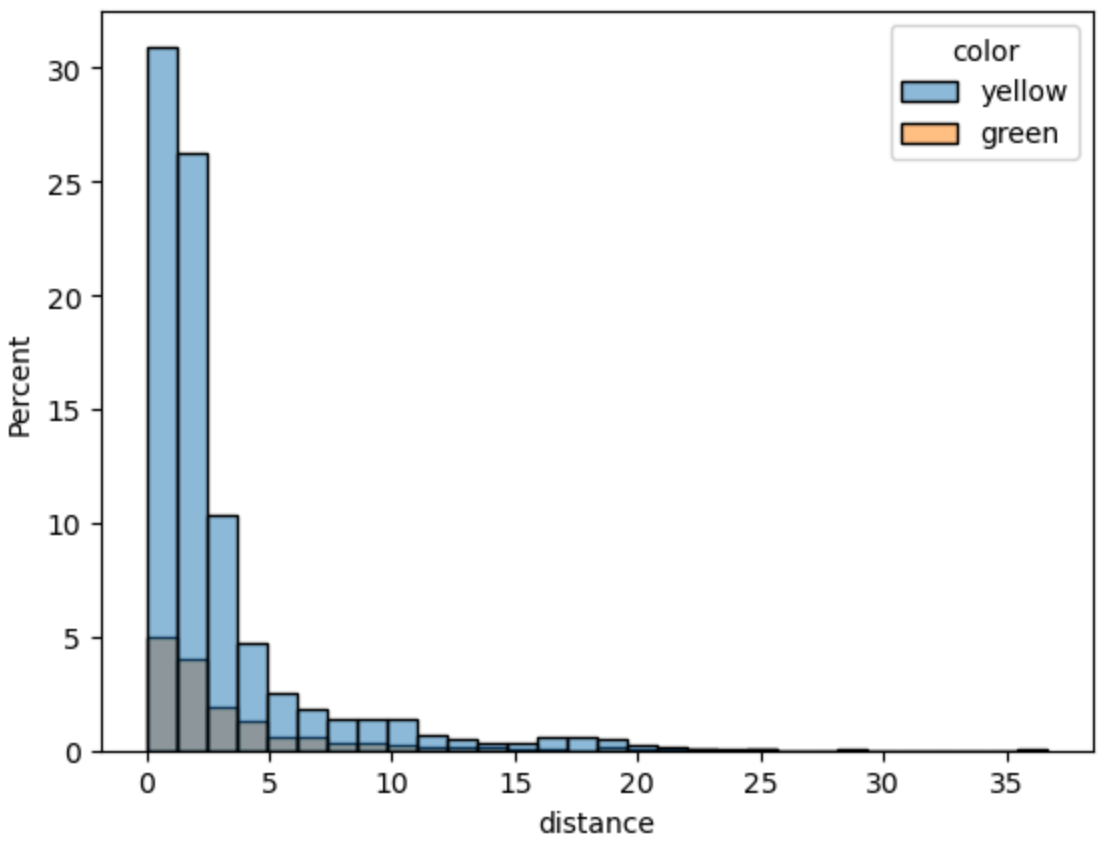
In [ ]:
```
#Cambio de count a Frequency en eje Y
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='frequency')
plt.show()
```



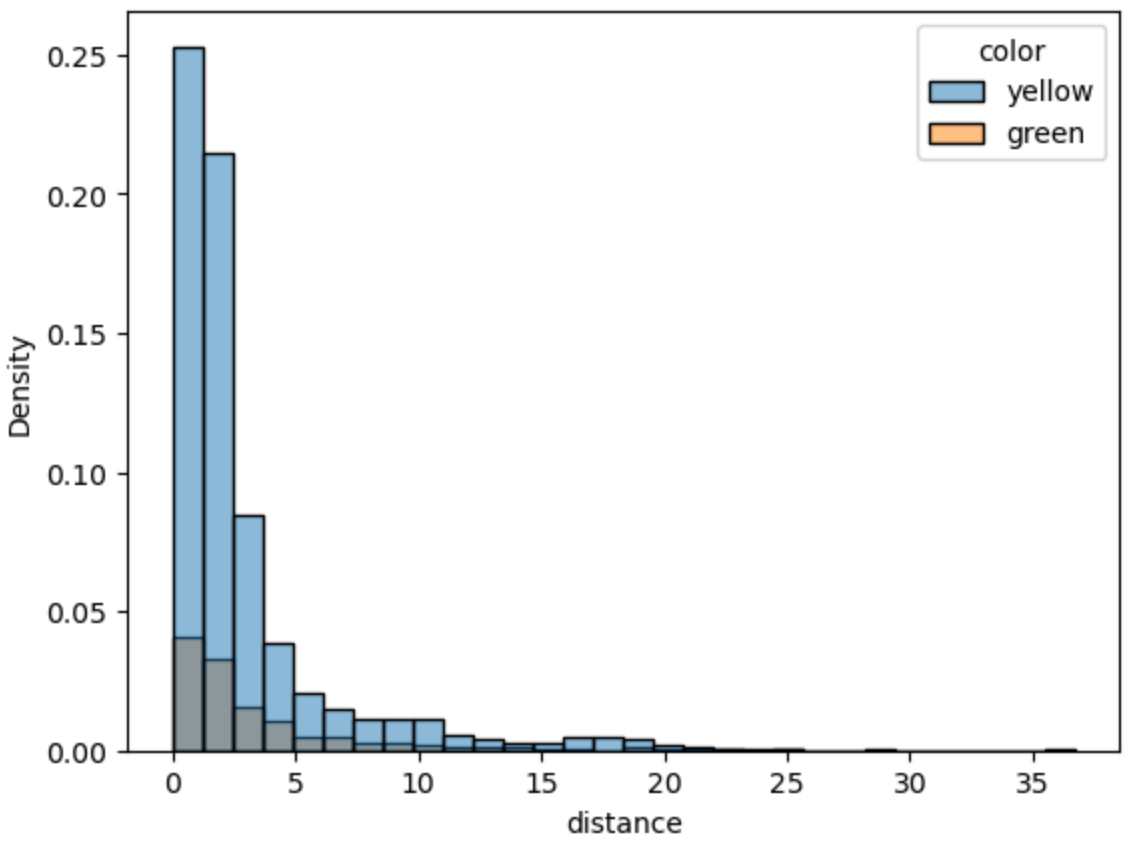
In [ ]:
```
#Trabajando con la probabilidad en Y
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='probability')
plt.show()
```



In [ ]:
```
#Trabajando con percent en Y
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='percent')
plt.show()
#Porcentajes
```

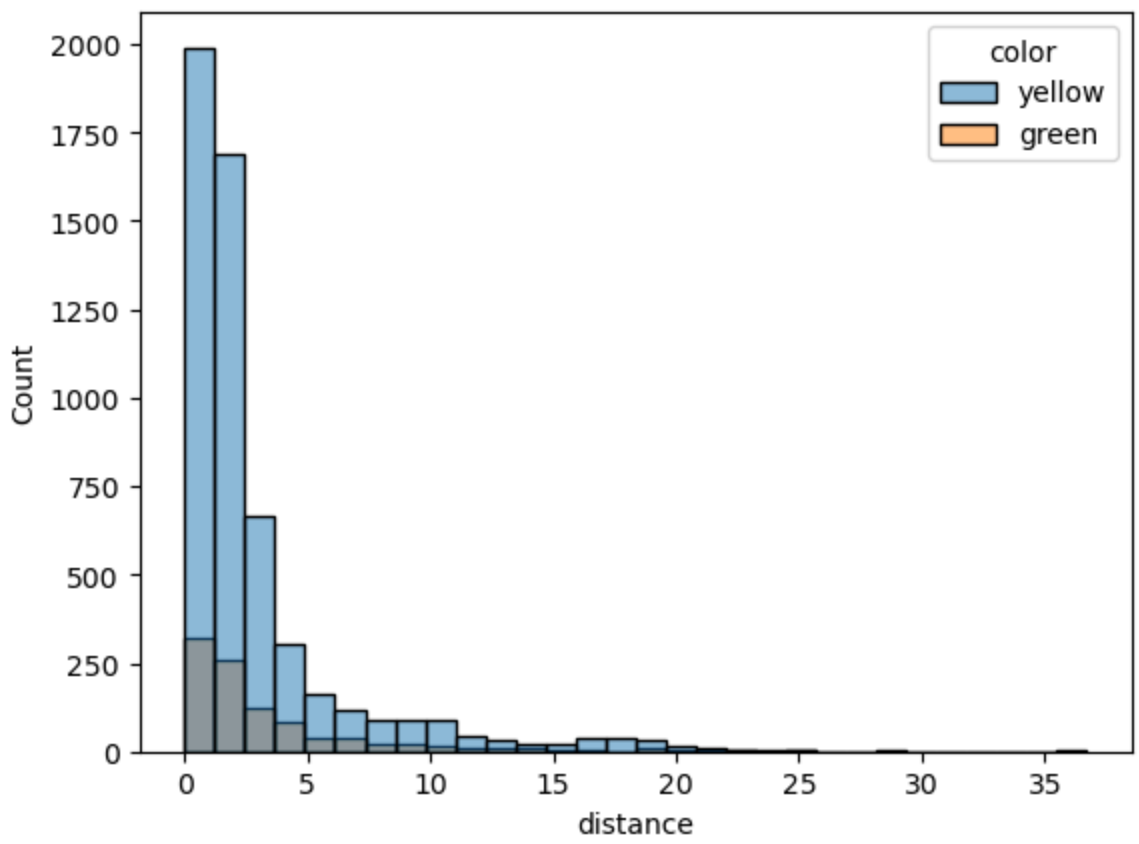


In [ ]:
```
#Trabajando con density en Y
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='density')
plt.show()
```



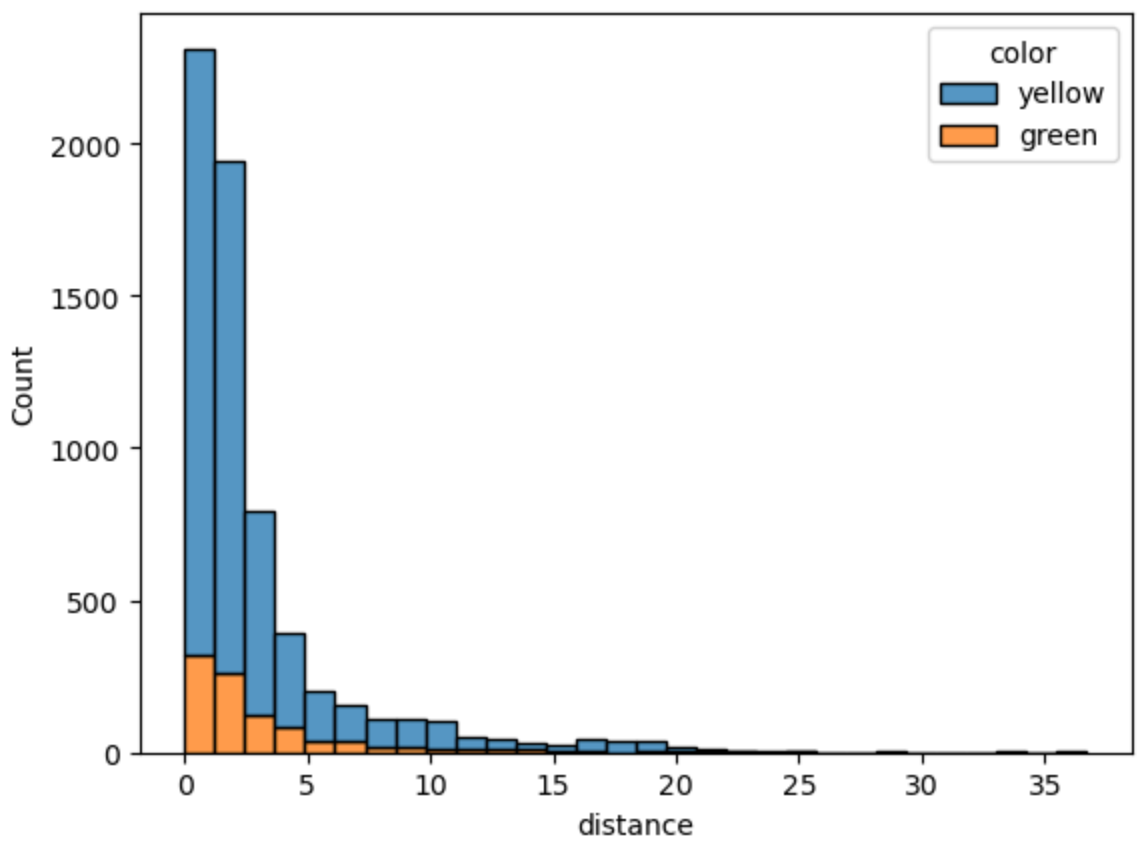
Count por default

In [ ]:
```
#Trabajando con count en Y
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='count')
plt.show()
```

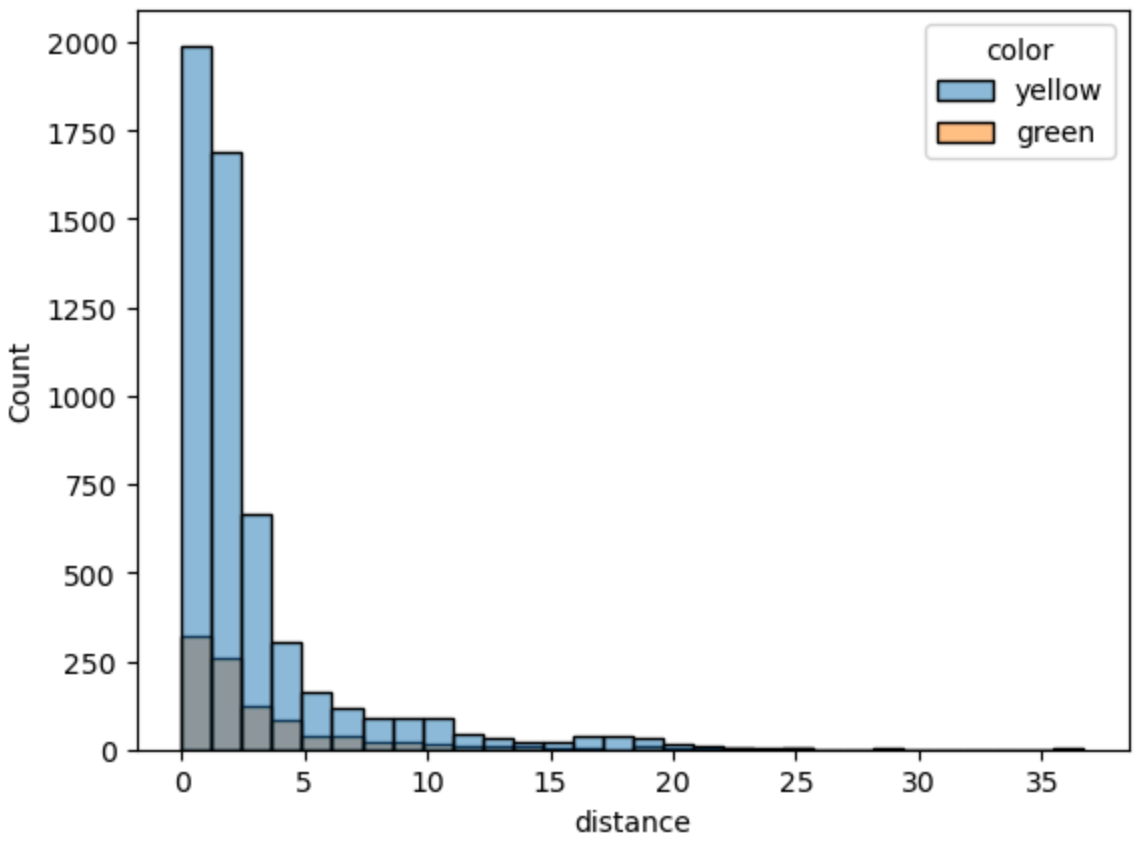


Parametro multiple

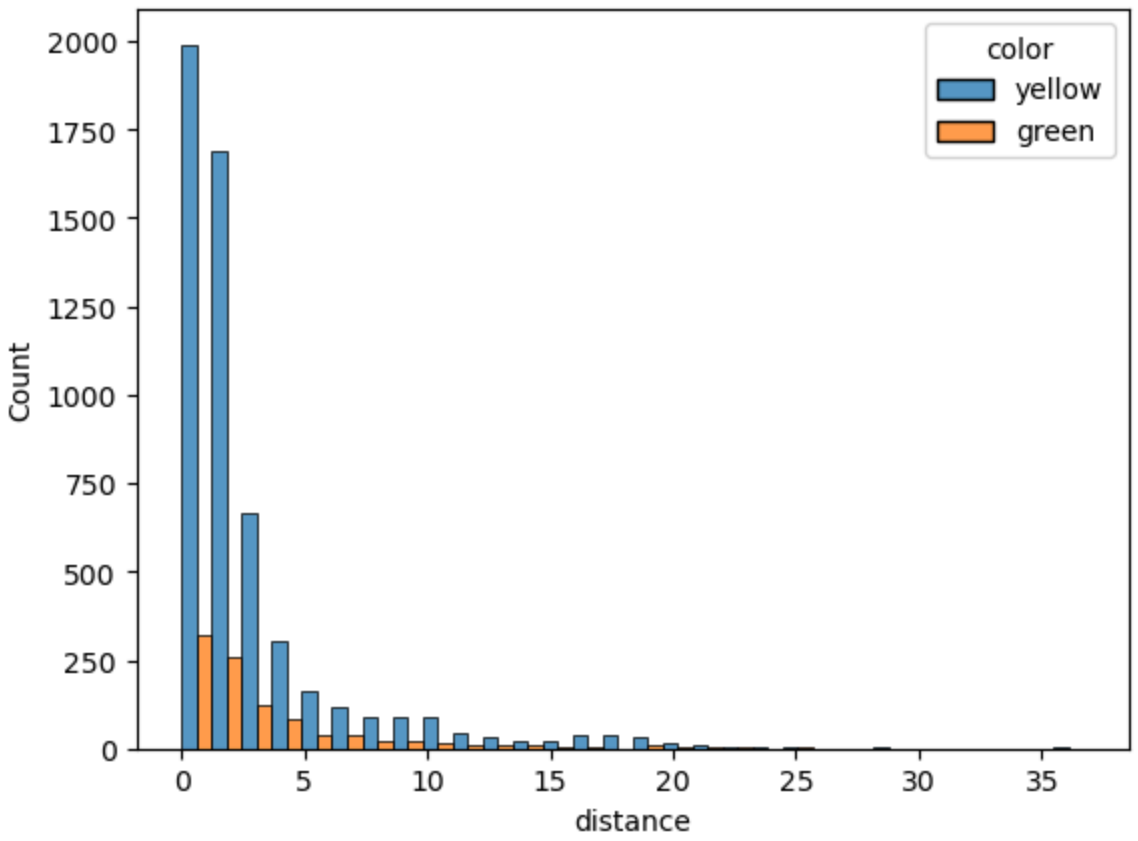
In [ ]:
```
#Trabajando con multiple = stack
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='count',multiple='stack')
plt.show()
```



In [ ]:
```
#Trabajando con multiple = layer
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='count',multiple='layer')
plt.show()
```



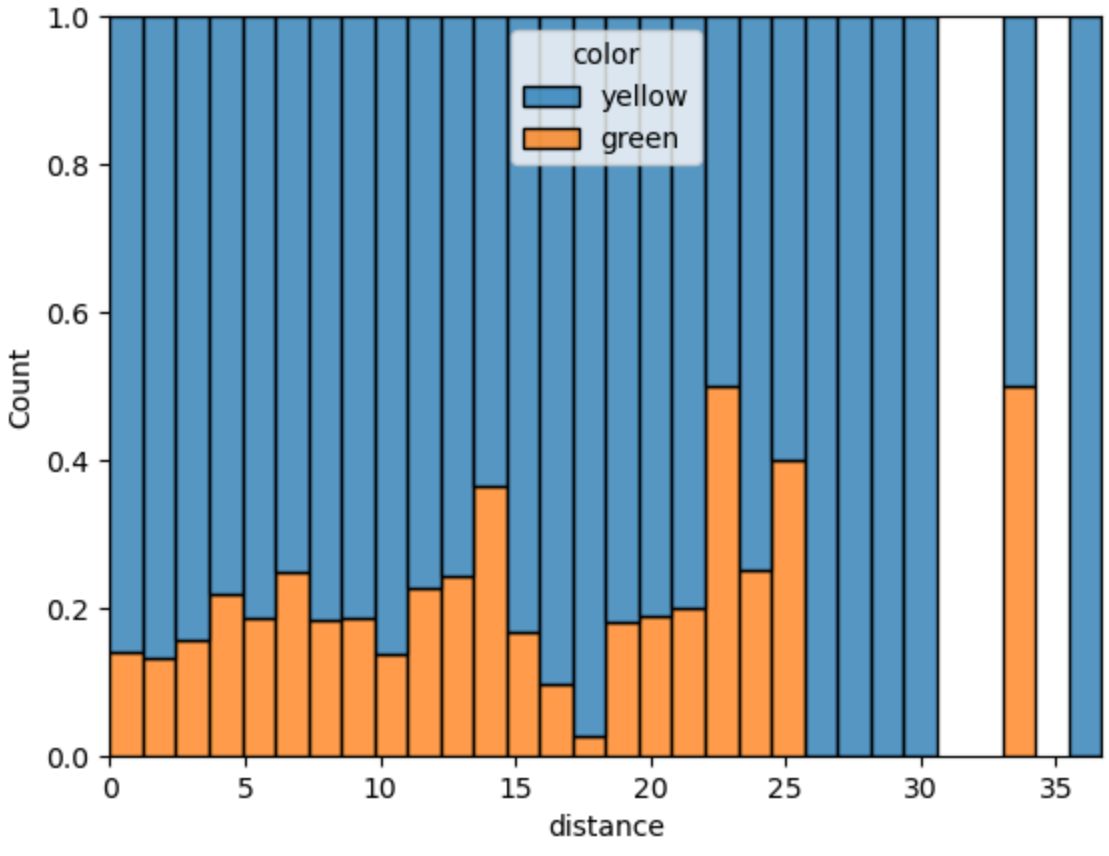
In [ ]:
```
#Trabajando con multiple = dodge
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='count',multiple='dodge')
plt.show()
```



Dodge:

Uno a lado del otro

In [ ]:
```
#Trabajando con multiple = fill
sns.histplot(data=taxis,x='distance',bins=30,cumulative=False,hue='color',stat='count',multiple='fill')
plt.show()
```

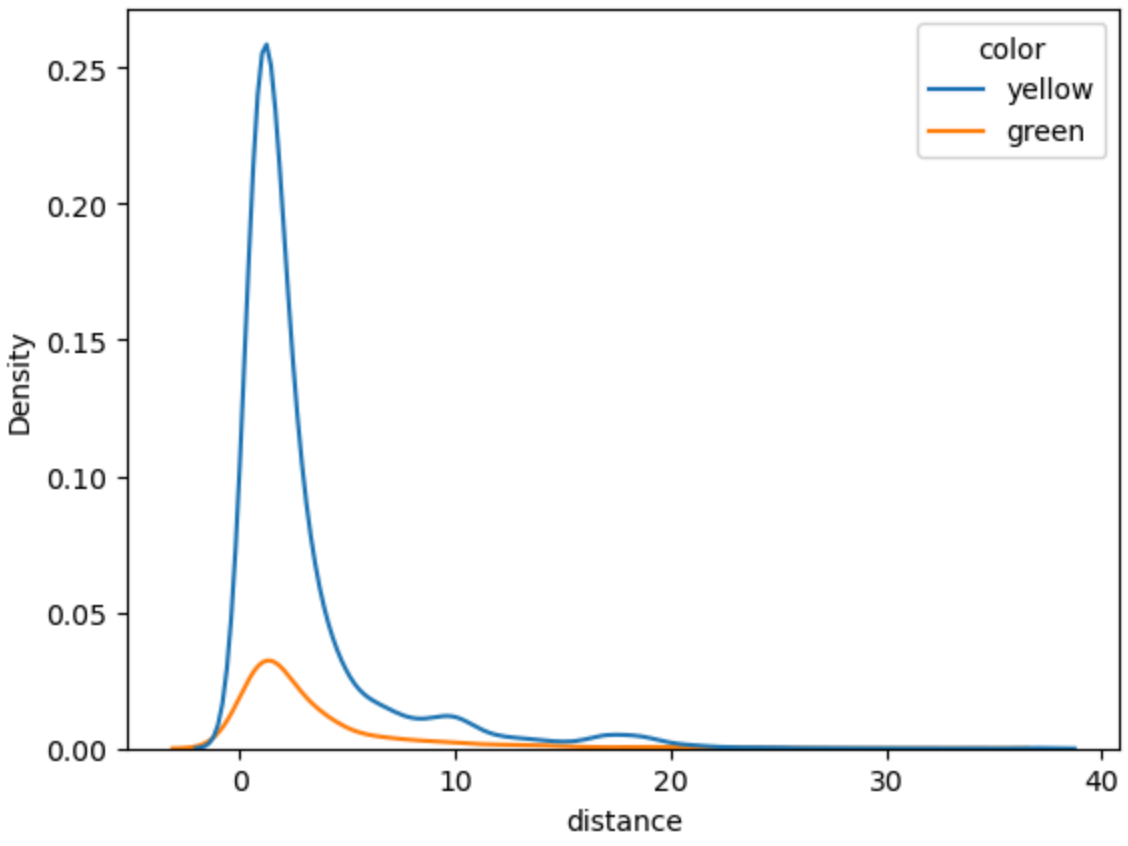


El dodge es muy bueno porque permite hacer comparativas de las distribuciones

KDE Diagrama de Densidad

In [ ]:
```
sns.kdeplot(data=taxis,x='distance',hue='color')
```

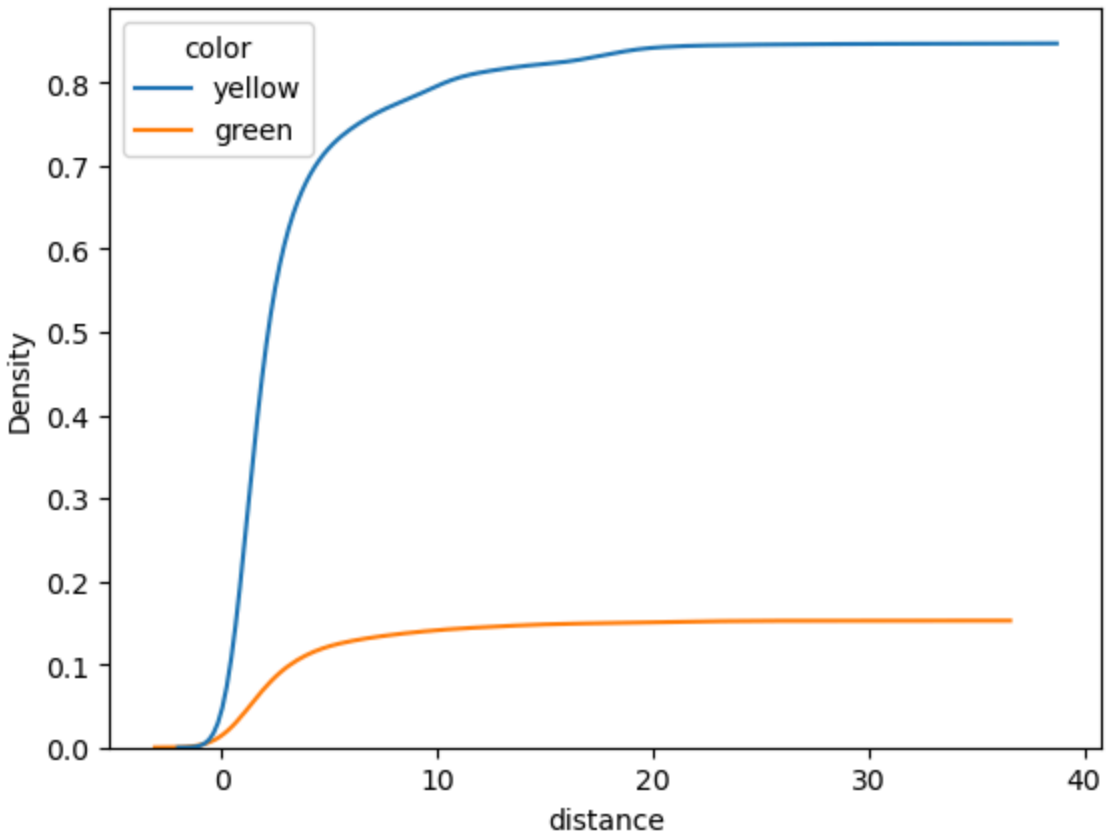
Out[ ]: <Axes: xlabel='distance', ylabel='Density'>



Este es diagrama de comportamiento de como se está coportando `distance`. Tanto por color `yellow` y por `green`.
Parece ser que hay mayor cantidad de taxis `yellow` que recorren mas distancia, y que la `distancia` más común recorrida está en un rango de 0 a 5

In [ ]:
```
#Acumulativo
sns.kdeplot(data=taxis,x='distance',hue='color',cumulative=True)
```

Out[ ]: <Axes: xlabel='distance', ylabel='Density'>



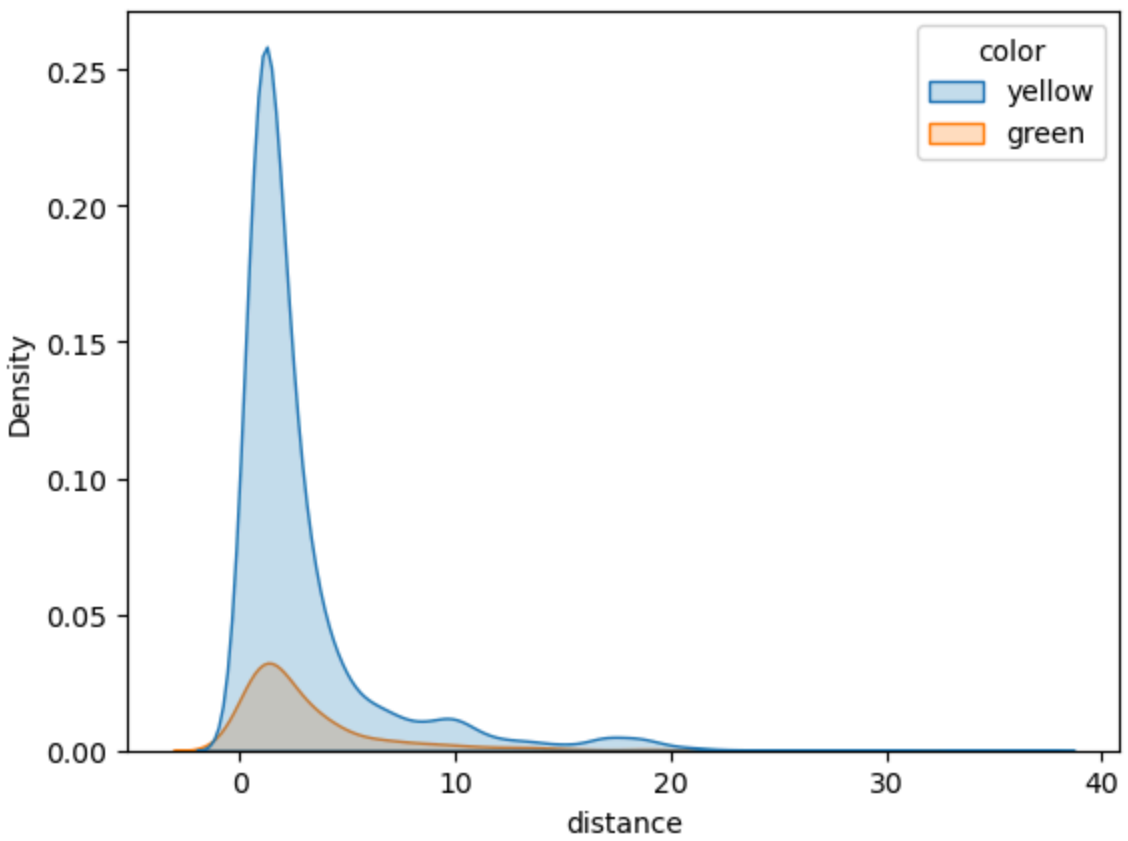
In [ ]:
```
sns.kdeplot(data=taxis,x='distance',hue='color',shade=True)
```

```
/tmp/ipykernel_35392/4017864630.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=taxis,x='distance',hue='color',shade=True)
```

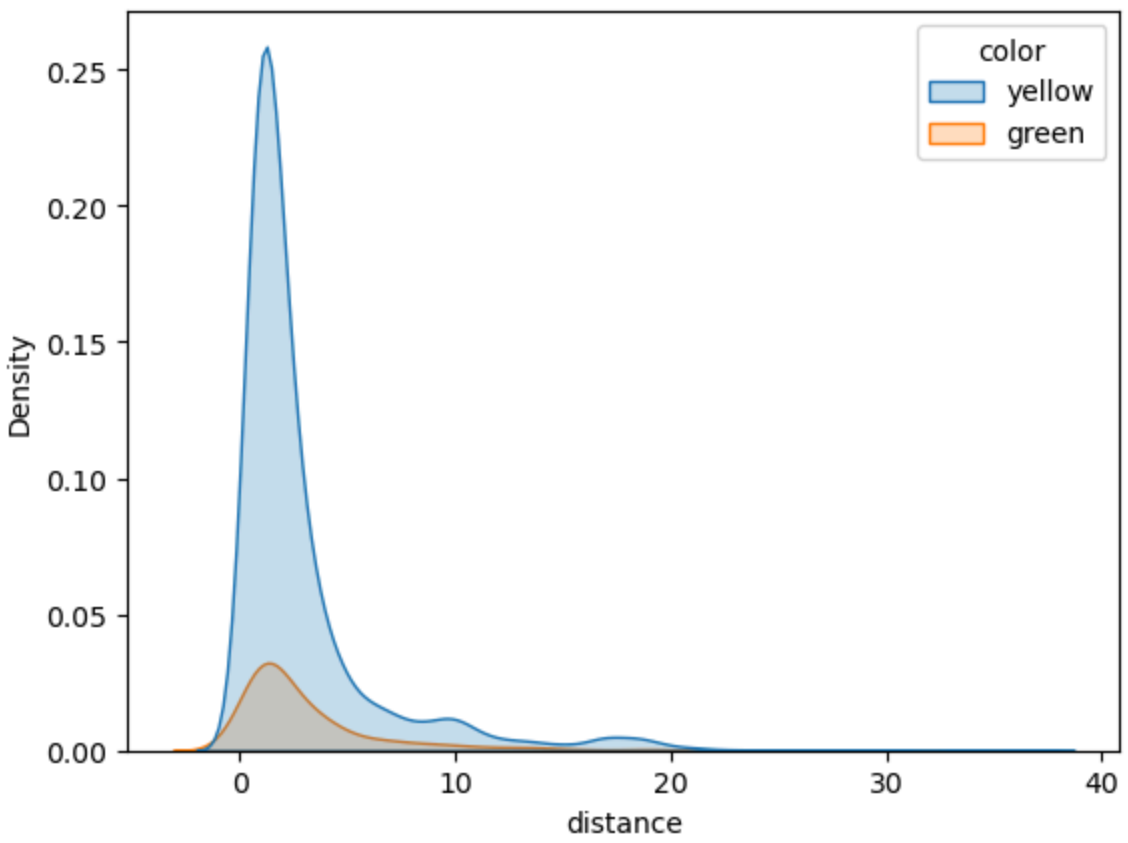
Out[ ]: <Axes: xlabel='distance', ylabel='Density'>



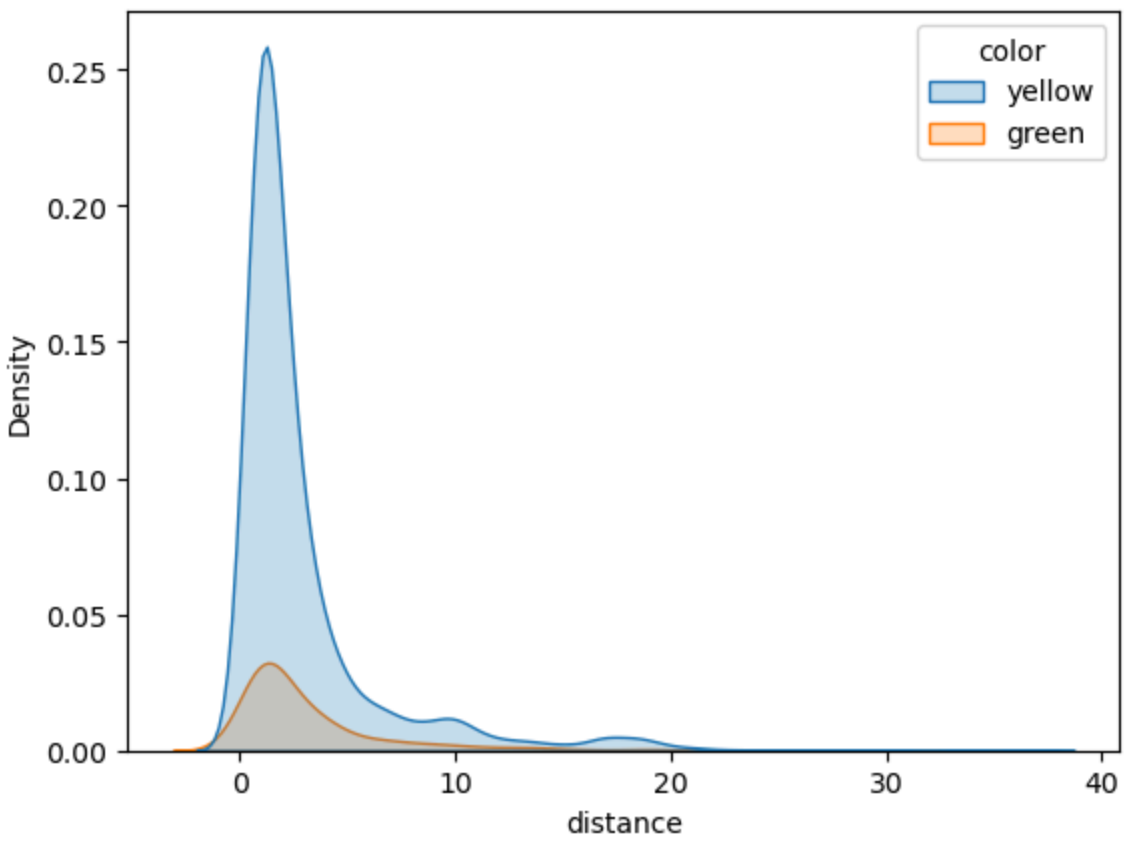
Shade

El parametro `shade` parece que ya está obsoleto y hay que cambiarlo por `fill`. Este parametro lo que hace es rellenar el área debajo de la curva

In [ ]:
```
sns.kdeplot(data=taxis,x='distance',hue='color',fill=True)
plt.show()
```

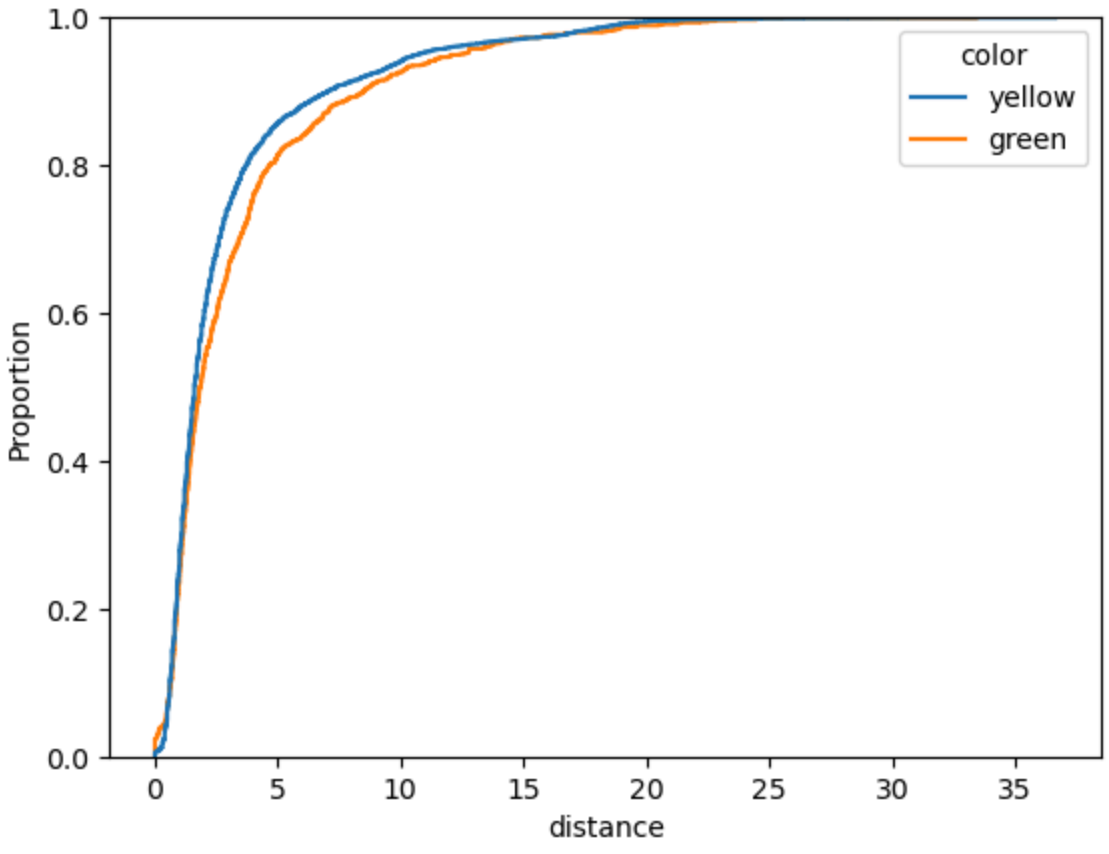


In [ ]:
```
#Adjustar la gráfica y los ejes automaticamente
#bw_adjust=1
sns.kdeplot(data=taxis,x='distance',hue='color',fill=True,bw_adjust=1)
plt.show()
```



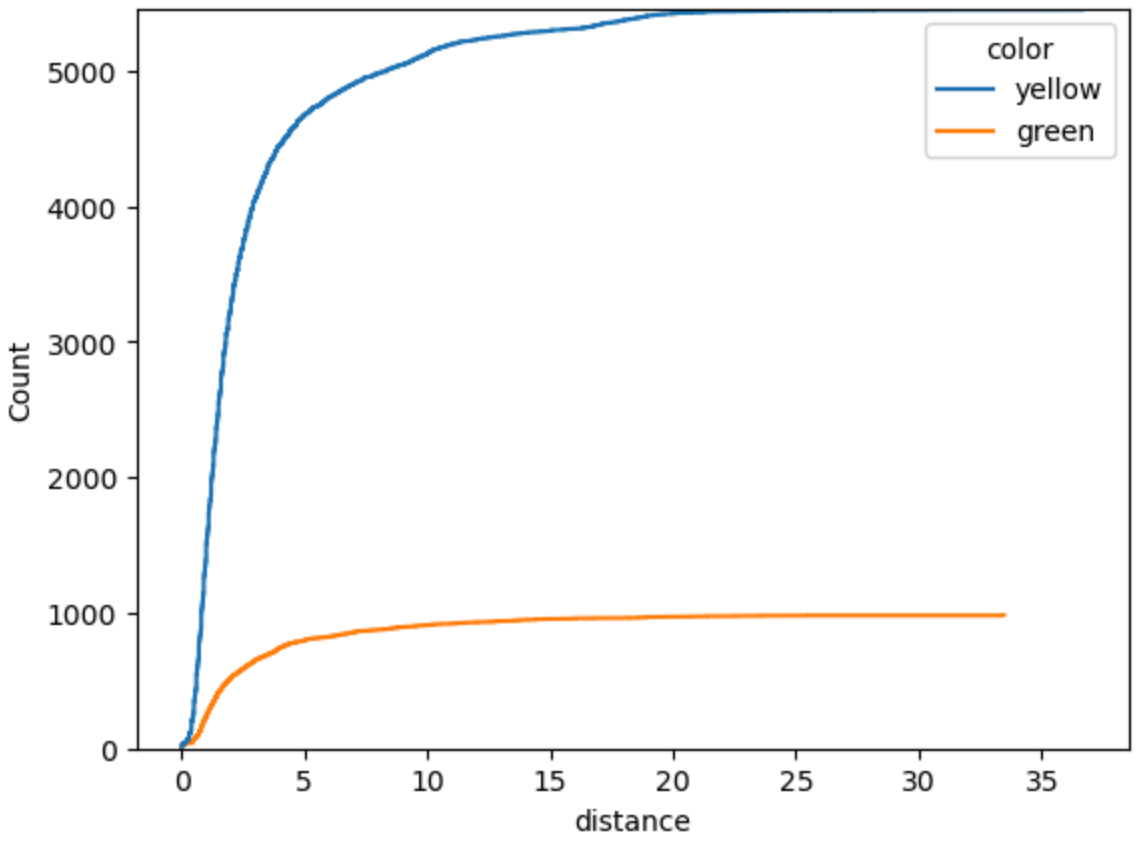
ecdfplot

In [ ]:
```
sns.ecdfplot(data=taxis,x='distance',hue='color')
plt.show()
```



Esta es una gráfica escalonada de la proporción que tienen las diferentes `distancias` a lo largo de la frecuencia, con respecto al color.

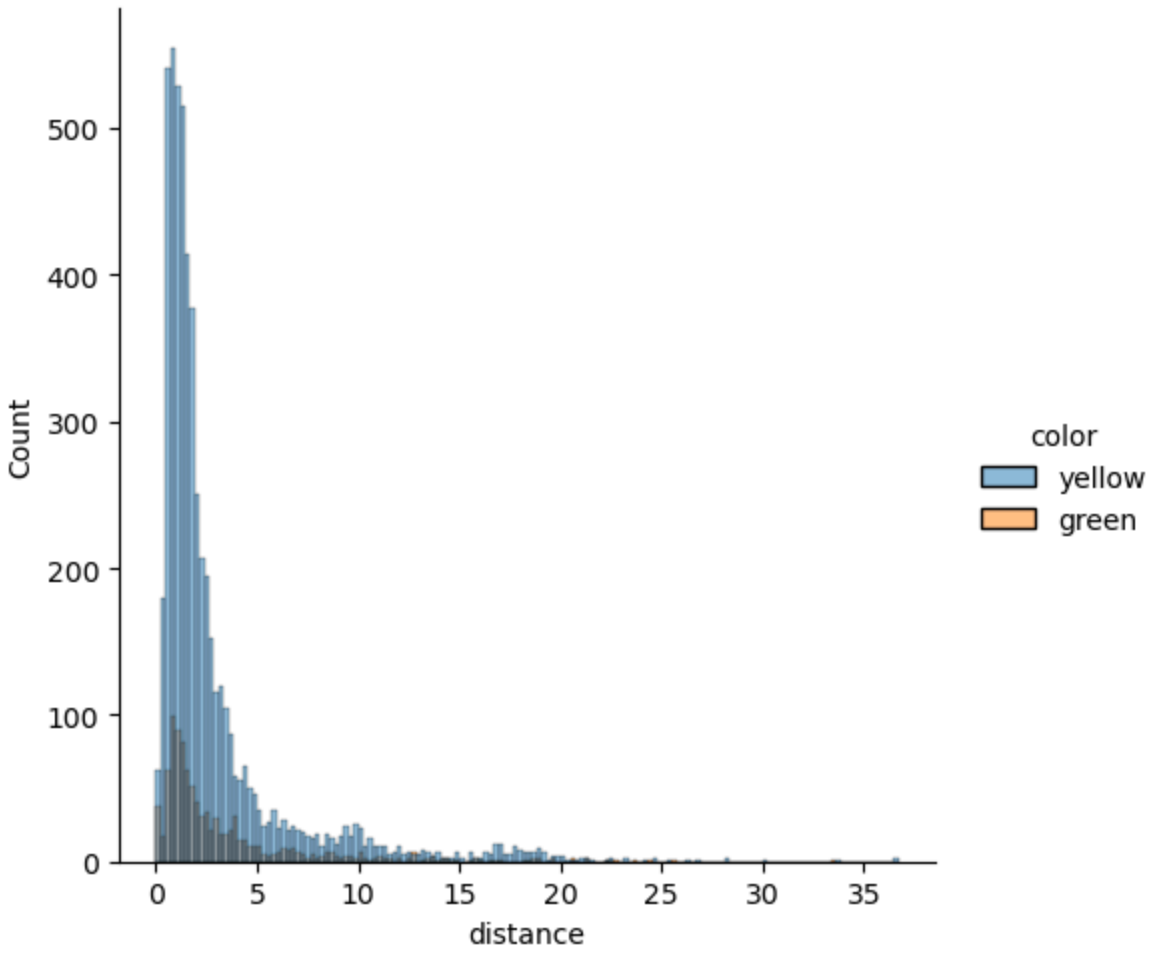
In [ ]:
```
#Stat= count
sns.ecdfplot(data=taxis,x='distance',hue='color',stat='count')
plt.show()
```



Esta es una de las grandes cosas de Seaborn

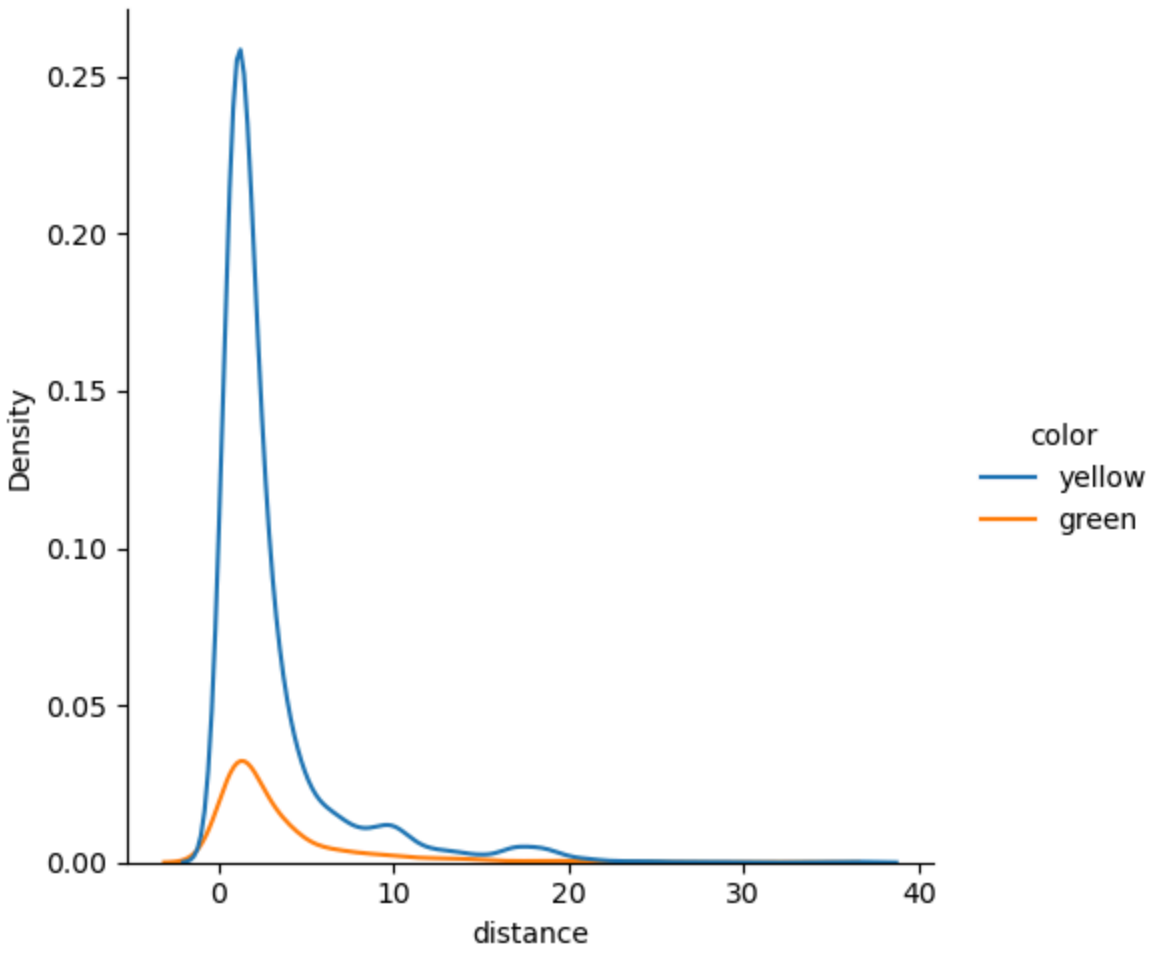
Diagrama de distribución

In [ ]:
```
sns.displot(data=taxis,x='distance',hue='color')
plt.show()
```

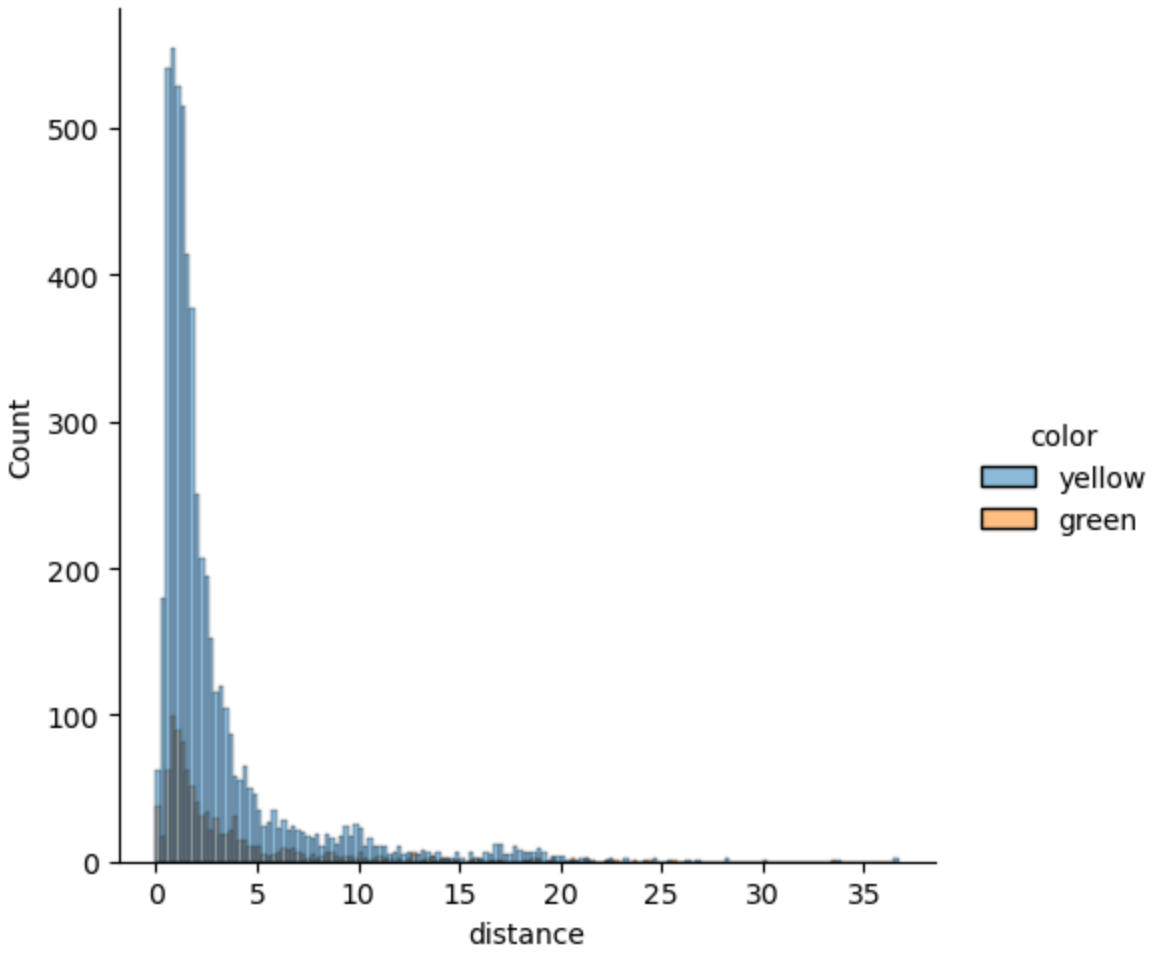


Me acaba de ejecutar un histograma

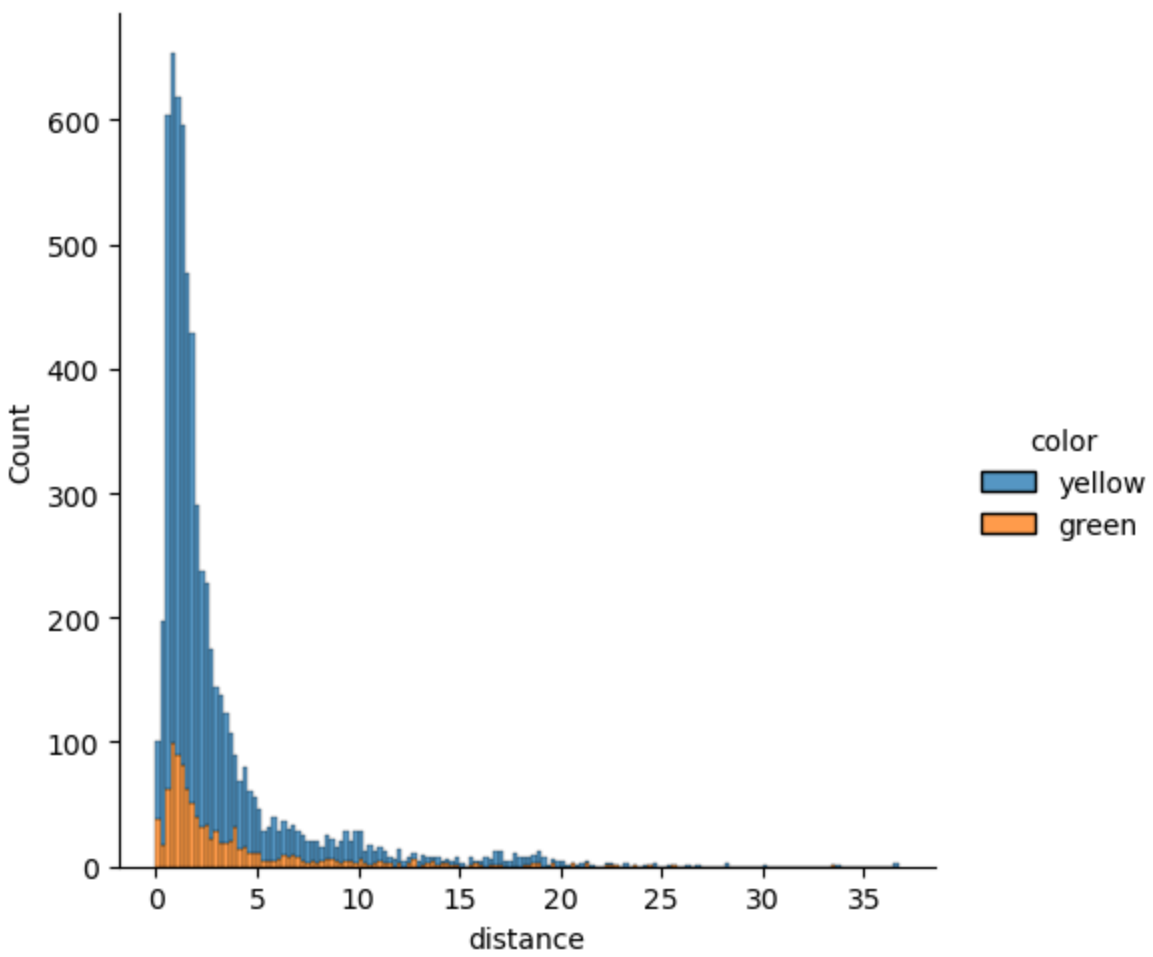
In [ ]:
```
#Puedo cambiar el tipo a KDE
#kind='kde'
sns.displot(data=taxis,x='distance',hue='color',kind='kde')
plt.show()
```



In [ ]:
```
#Histograma
#kind='hist'
sns.displot(data=taxis,x='distance',hue='color',kind='hist')
plt.show()
```



In [ ]:
```
#Multiple stack
#kind='hist'
sns.displot(data=taxis,x='distance',hue='color',kind='hist',multiple='stack')
plt.show()
```

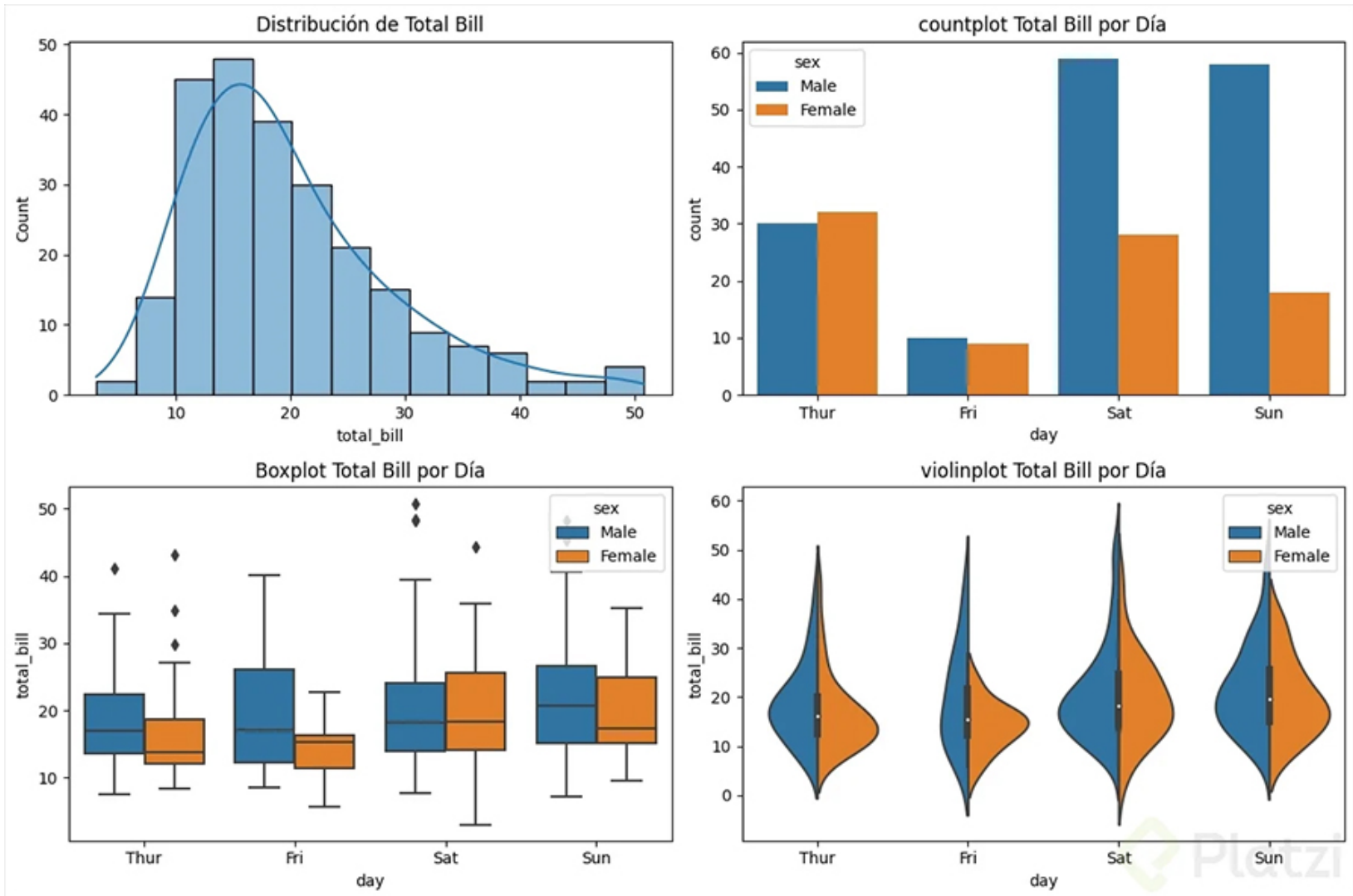


IMPORTANTE:

No te compliques demasiado y ajusta los parametros como en las gráficas anterioresm pero este tipo de gráfico, te ofrece una gran ventaja con `kind` y puedes ajustarlo a lo que trabajes para graficos de distribución. Simplemente especificandolo en `kind`

Aporte por Camilo Gonzalo Morales Sanchez

Se explica por partes como agregar mas de un grafica con seaborn, aplicando lo visto en clases anteriores




```
#importar librerias
import seaborn as sns
import matplotlib.pyplot as plt
#cargar la data
tips = sns.load_dataset('tips')
#subplots
fig, axes = plt.subplots(nrows=2,ncols=2, figsize = (12,8))
#seaborn en cada subsplots

# Subplot 1: Distribución de la cuenta total
sns.histplot(data=tips, x="total_bill", kde=True, ax=axes[0, 0])
axes[0, 0].set_title("Distribución de Total Bill")

# Subplot 2: cuenta por cada dia y cada sexo
sns.countplot(data = tips, x = 'day', hue = 'sex', ax=axes[0,1])
axes[0, 1].set_title("countplot Total Bill por Día")

# Subplot 3:  Boxplot de la cuenta total vs día de la semana por cada sexo
sns.boxplot(data = tips, x= 'day', y='total_bill', hue='sex', dodge=True, ax=axes[1,0])
axes[1, 0].set_title("Boxplot Total Bill por Día")

# Subplot 4: violinplot de la cuenta total vs el dia de la semana por cada sexo
sns.violinplot(data = tips, x= 'day', y = 'total_bill', hue='sex',split= True, dodge= True,ax=axes[1,1])
axes[1, 1].set_title("violinplot Total Bill por Día")

# Ajustar espaciado entre subplots
plt.tight_layout()

# Mostrar los subplots
plt.show()
```

Referencias:

- Visualizing distributions of data