

Ejemplo de uso para la biblioteca paleta_brewer

April 27, 2020

Cisneros Aguilar Sara Kenia
Diaz Moreno Rodrigo

Al momento de realizar la presentación de nuestra información, siempre sale la pregunta, ¿cuales son los colores adecuados para presentarla?, y es que no todos son adecuados para los distintos tipos de gráficas, además debemos tomar mucho en cuenta nuestros objetivos, pues si queremos resaltar una cierta figura de un conjunto, podría ser muy útil una escala de grises para no perder la atención entre los diferentes colores que utilizamos, pues, si bien es cierto que los colores son muy llamativos para atrapar al publico, también podemos perder su atención.

Para esto, se han desarrollado diferentes tipos de paletas, y maneras de ver el color, por ejemplo, con el sistema RGB utilizamos tres componentes para un color, el rojo, el verde y el azul, pero, si nosotros vemos un rosa pálido, ¿podemos descomponerlo en estos componentes?, no es muy intuitivo, por lo que este sistema no es el mas adecuado, para mejorar estas fallas, se ha recurrido a sistemas diferentes como el HSV, que define un color en tono, saturación y valor, así es un poco mas fácil descomponer un color, por ejemplo un azul oscuro intenso, de esta manera nos damos una idea del tono, la saturación y el valor.

Desafortunadamente HSV tiene un problema, no es preceptivamente uniforme, que quiere decir esto?, que defina los colores tal cual los percibimos nosotros, por ejemplo si imaginamos un espacio continuo donde se encuentran los colores, los diferentes tonos serán proporcionales a sus distancias, no vamos a dar un salto drástico de blanco a negro, ni de amarillo a negro.

En HSV recorrer algún valor puede tener un cambio drástico, entonces cual es la solución?, lo ideal sería contar con una paleta de colores que se base en la percepción y utilice los parámetros intuitivos de HSV, este espacio es LCH, este define un color en función del brillo percibido, su pureza y el tono, el problema es que aún no existe un software que nos ayude a darle un buen uso y no se puede aprovechar bien.

Entonces una alternativa son las paletas Brewer, creadas por Cynthia Brewer, con el propósito de la cartografía, pero se les ha encontrado uso en otros campos.

Cynthia Brewer pensaba que no se pueden mostrar más de tres tipos de datos con el color, la confusión se establece en la cuarta variable, pero se pueden mostrar con éxito tres variables en un mapa. En los datos topográficos, por ejemplo, se podrían representar de la siguiente manera para que el mapa tome forma

1. los tonos de color podrían mostrar la dirección de las pendientes
2. la saturación de los colores podrían indicar la inclinación de la pendiente
3. la claridad representa la tercera variable, la iluminación

Con esto en mente, creó una herramienta para representar de una manera adecuada la información de un mapa. El objetivo de estas paletas es ayudar con la buena presentación de la

información tomando en cuenta las diferentes formas de representarla y a los diferentes tipos de vista, pues se considera a las personas que sufren de daltonismo. Generalmente cometemos el error de suponer que todos perciben los colores de la misma manera, ¿alguna vez pensaste en ellos al momento de hacer uso del color?

Si generamos una gráfica donde tenemos un conjunto de datos categóricos con orden y para distinguirlos asignamos diferentes tonalidades de tal manera que los tonos más fuertes son los que tienen mas valor, podríamos pensar que todos agrupan las distintas relaciones de la misma manera, pero no es así, el 8% de los hombres y menos del 1% de las mujeres sufren daltonismo, pero en general el 50% de la población tiene al menos un grado de daltonismo, por lo que la mitad de nuestro publico no podría relacionar bien los conjuntos de información.

Primero veamos los tres tipos básicos de paletas brewer y sus respectivos usos:

1. Cualitativo: conjunto de colores sin orden, utilizados para gráficos de barras y gráficos circulares que representan categorías
2. Secuencial: conjunto de colores con orden de diferencia uniforme entre ellos, es decir son proporcionales a la distancia que existe entre ellos, utilizados para representar mapas de color o con alguna secuencia existente
3. Divergente: conjunto de colores secuenciales en dos direcciones que parten de un mismo color, utilizados para representar mapas de color o con alguna secuencia existente
4. Escala de grises (todos los tipos anteriores están en esta presentación): utilizados para la separación de figuras, clases, etc. en tonos de gris, blanco y negro

Tipos de daltonismo considerados:

1. Deuteranopia
2. Protanopia
3. Tritanopia

Para facilitar el uso de estas paletas se ha generado una biblioteca que nos permite elegir el tipo, ya sea secuencial, divergente o cualitativo, dándonos la opción de configurar la paleta. De la misma manera nos proporciona la facilidad para las paletas dirigidas a personas con discapacidad visual como es el daltonismo.

0.1 Ejemplos de su uso de la biblioteca `paleta_brewer`

Primero importamos las bibliotecas necesarias:

```
[1]: import paleta_brewer
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

0.1.1 Paleta tipo Divergente

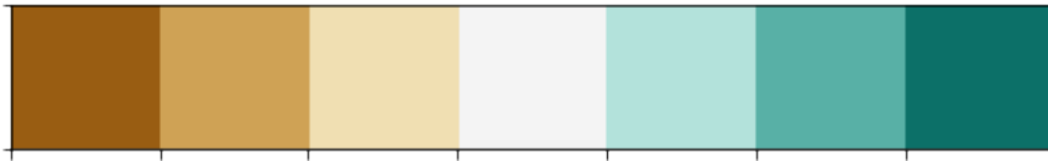
Navegamos dentro de la interfaz a la primera opción, la cual es “Paletas Normales” y dentro de ella, escogemos de nuevo la opción 1, que es “Paleta Divergente”.

Continuamos escogiendo la primera opción llamada “Escoger paleta”. Esta opción nos desplegará algunas paletas que están en nuestro programa y a continuación el usuario escogerá cuál es la que más se adecue a sus necesidades.

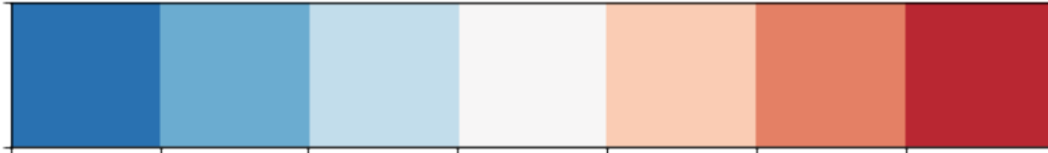
```
[2]: paleta_brewer.run()
```

<Figure size 432x288 with 0 Axes>

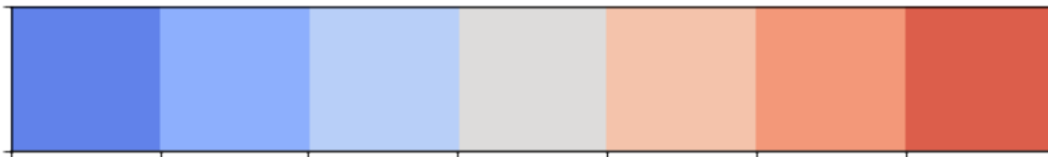
1) Paleta BrBG



2) Paleta RdBu_r



3) Paleta coolwarm



4) Paleta YlGnBu



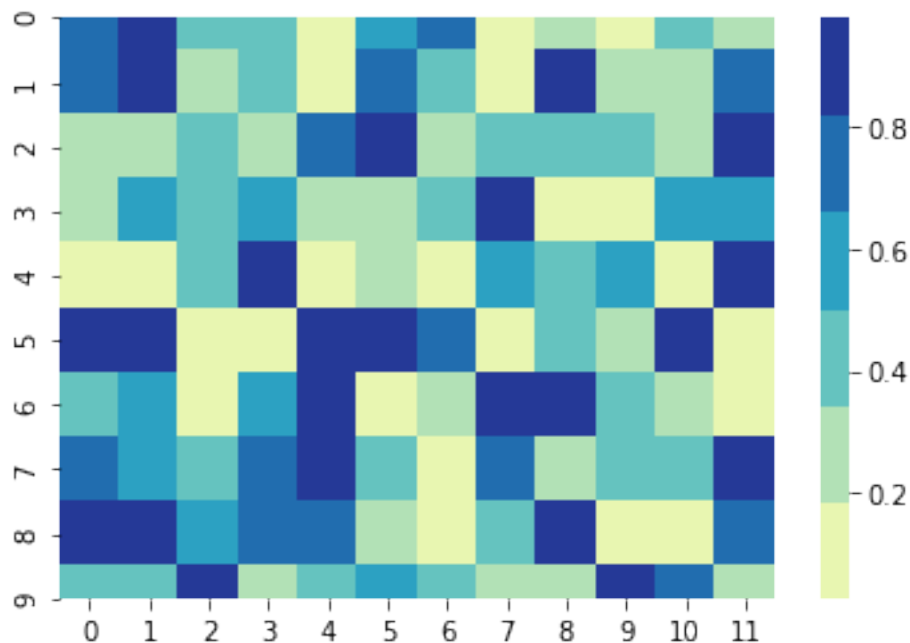
Ingresa el numero de la opcion: 4

Paleta Seleccionada

¡Vuelva pronto!

Para probar la paleta escogida, graficámos un heatmap de números aleatorios uniformemente distribuidos. La cual es una gráfica que se adecua a la paleta de tipo divergente.

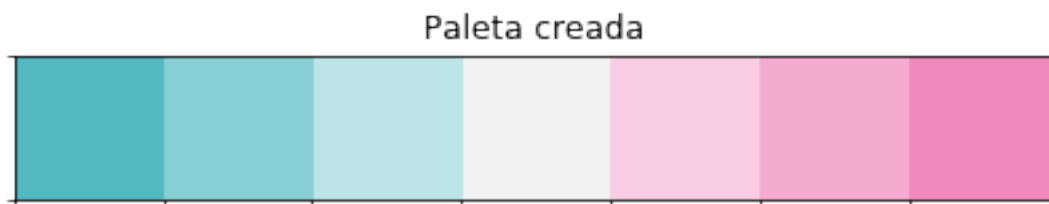
```
[3]: uniform_data = np.random.rand(10, 12)
ax = sns.heatmap(uniform_data, cmap=sns.color_palette())
```



Ahora probamos la opcion 2 de las paletas divergentes, la cual es “Crear paleta”. Esta nos permite hacer nuestra propia paleta divergente, ingresando los datos que estan en el sistema de colores hus1.

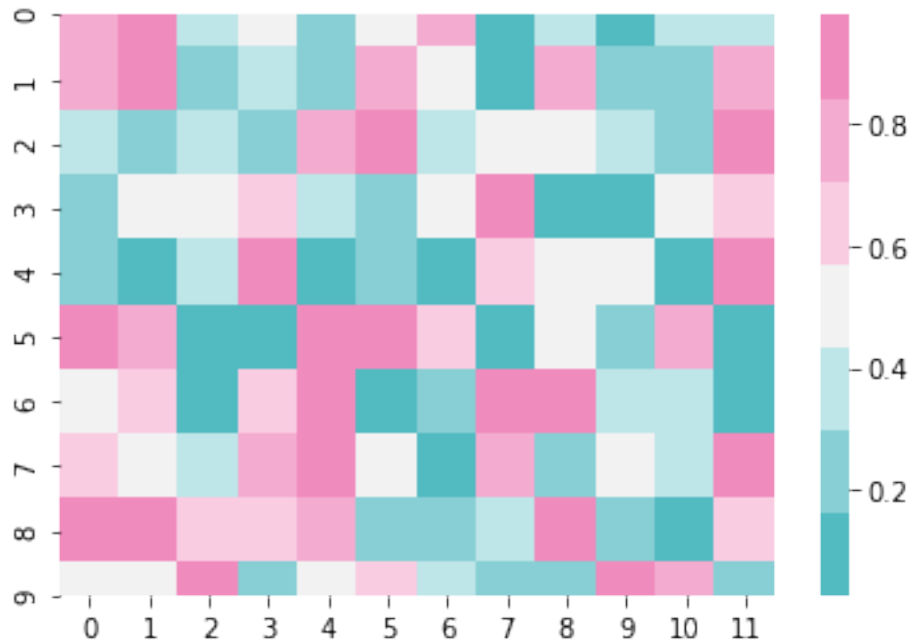
```
[5]: paleta_brewer.run()
```

Ingrese el grado del color 1: 200
 Ingrese el grado del color 2: 340
 Ingrese la saturacion: 80
 Ingrese la luminocidad: 70
 Paleta Seleccionada
 ¿Vuelva pronto!



Nuevamente probamos esa paleta con un heatmap.

```
[6]: ax = sns.heatmap(uniform_data, cmap=sns.color_palette())
```

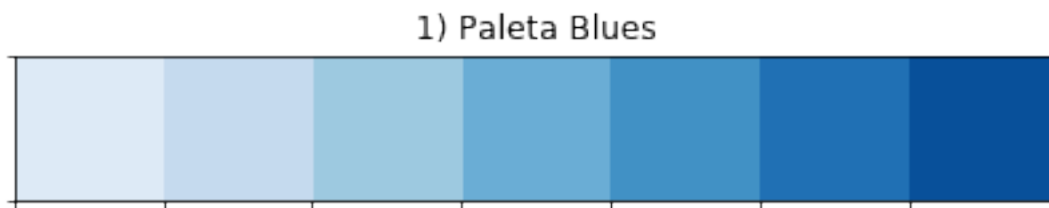


0.1.2 Paleta tipo Secuencial

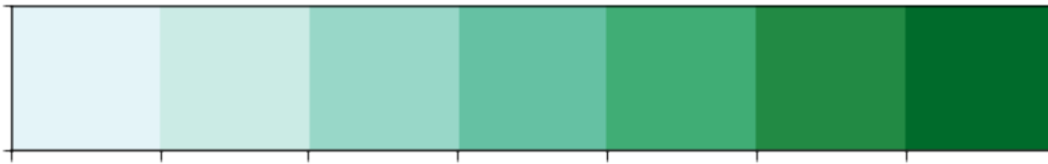
Navegamos dentro de la interfaz a la primera opción, la cual es “Paletas Normales” y dentro de ella, escogemos la opción 2, que es “Paleta Secuencial”. Continuamos escogiendo la primera opción llamada “Escoger paleta”, desplegando algunas paletas que están en nuestro programa y a continuación el usuario escogerá cuál es la que más se adecue a sus necesidades.

```
[7]: paleta_brewer.run()
```

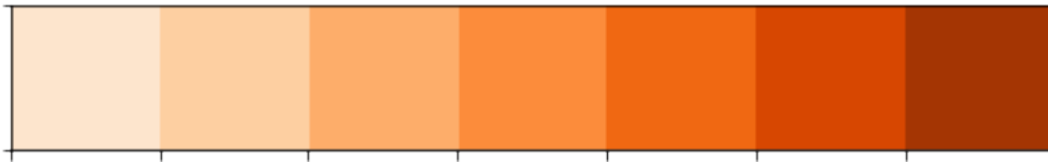
<Figure size 432x288 with 0 Axes>



2) Paleta BuGn



3) Paleta Oranges



4) Paleta Greys

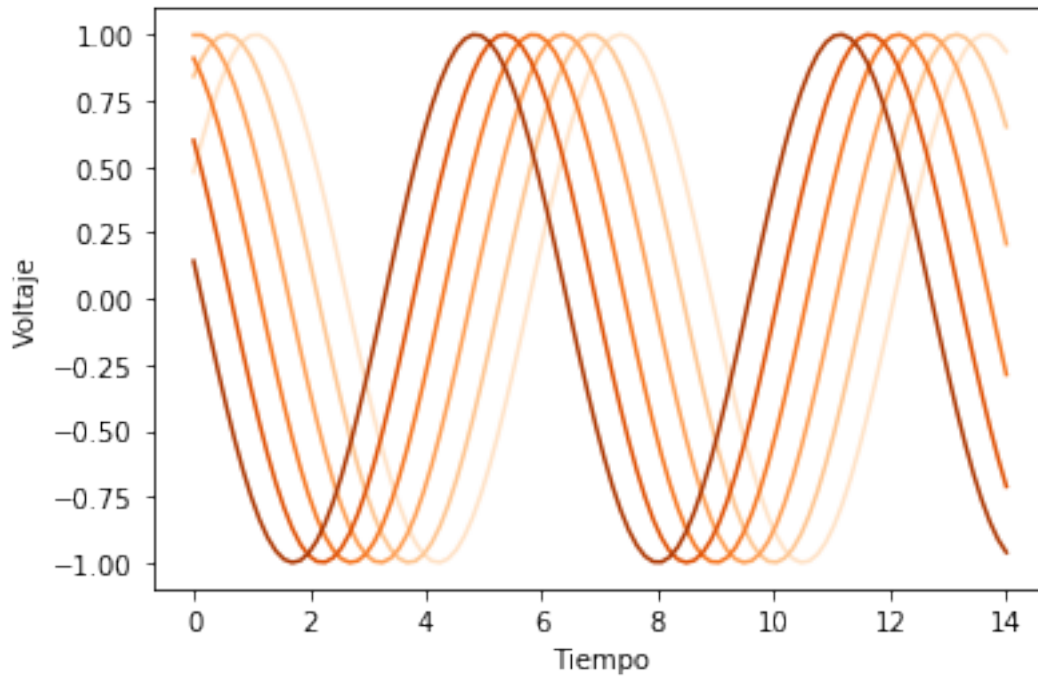


Ingresa el numero de la opcion: 3
Paleta Seleccionada
¡Vuelva pronto!

Probamos la paleta seleccionada con una funcion que despliega ondas sinusoidales que estan desplazadas. Las cuales al tener un orden (que seria el tiempo), nos conviene ocupar una paleta secuencial.

```
[38]: def sinplot():  
      x = np.linspace(0, 14, 100)  
      for i in range(1, 7):  
          plt.plot(x, np.sin(x+(i/2)))  
          plt.xlabel("Tiempo")  
          plt.ylabel("Voltaje")
```

```
[39]: sinplot()
```



Ahora probamos la opcion 2 de las paletas secuenciales, la cual es “Crear paleta”. Esta nos permite hacer nuestra propia paleta secuencial, ingresando los datos que estan en el sistema de colores hus1.

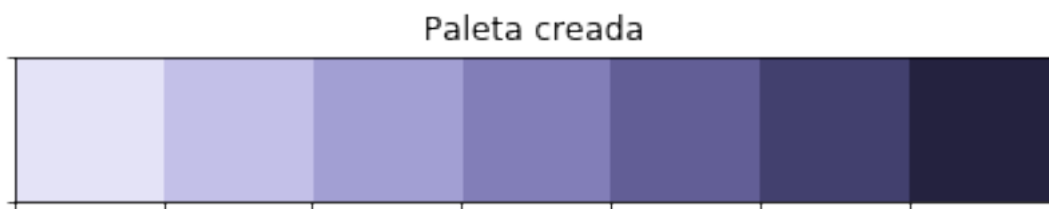
```
[41]: paleta_brewer.run()
```

Ingrese el grado del color: 81

Ingrese la luminocidad: 90

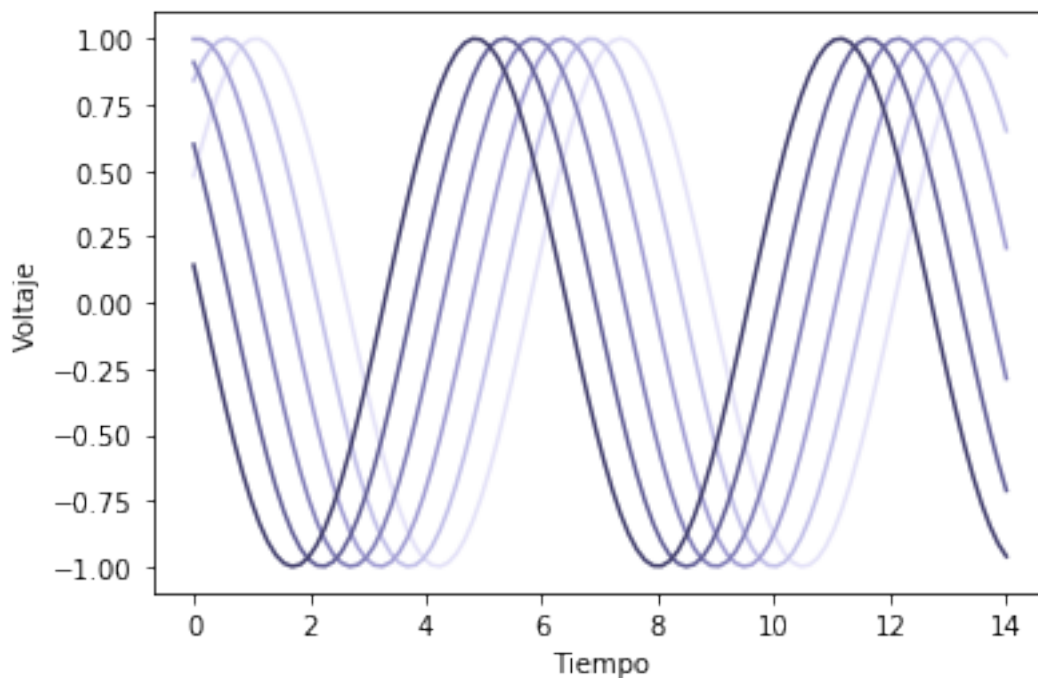
Paleta Seleccionada

¡Vuelva pronto!



Probamos la paleta creada

```
[42]: sinplot()
```

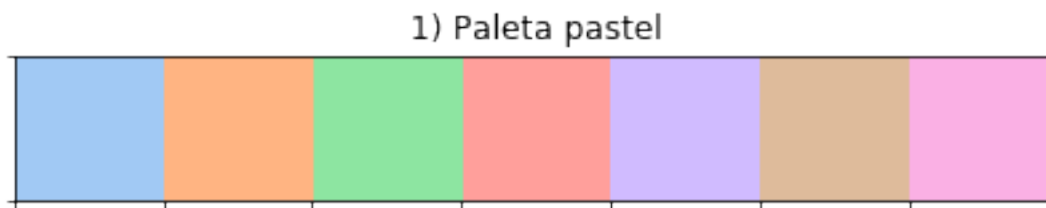


0.1.3 Paleta tipo Cualitativa

Navegamos dentro de la interfaz a la primera opción, la cual es “Paletas Normales” y dentro de ella, escogemos la opción 3, que es “Paleta Cualitativa”. Continuamos escogiendo la primera opción llamada “Escoger paleta”. Esta opción nos desplegará algunas paletas que están en nuestro programa y a continuación el usuario escogerá cuál es la que más se adecue a sus necesidades.

[43]: `paleta_brewer.run()`

<Figure size 432x288 with 0 Axes>



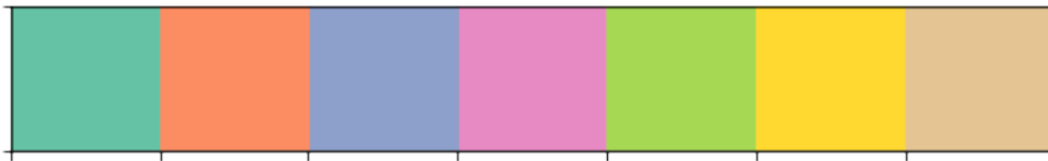
2) Paleta Paired



3) Paleta Set1



4) Paleta Set2



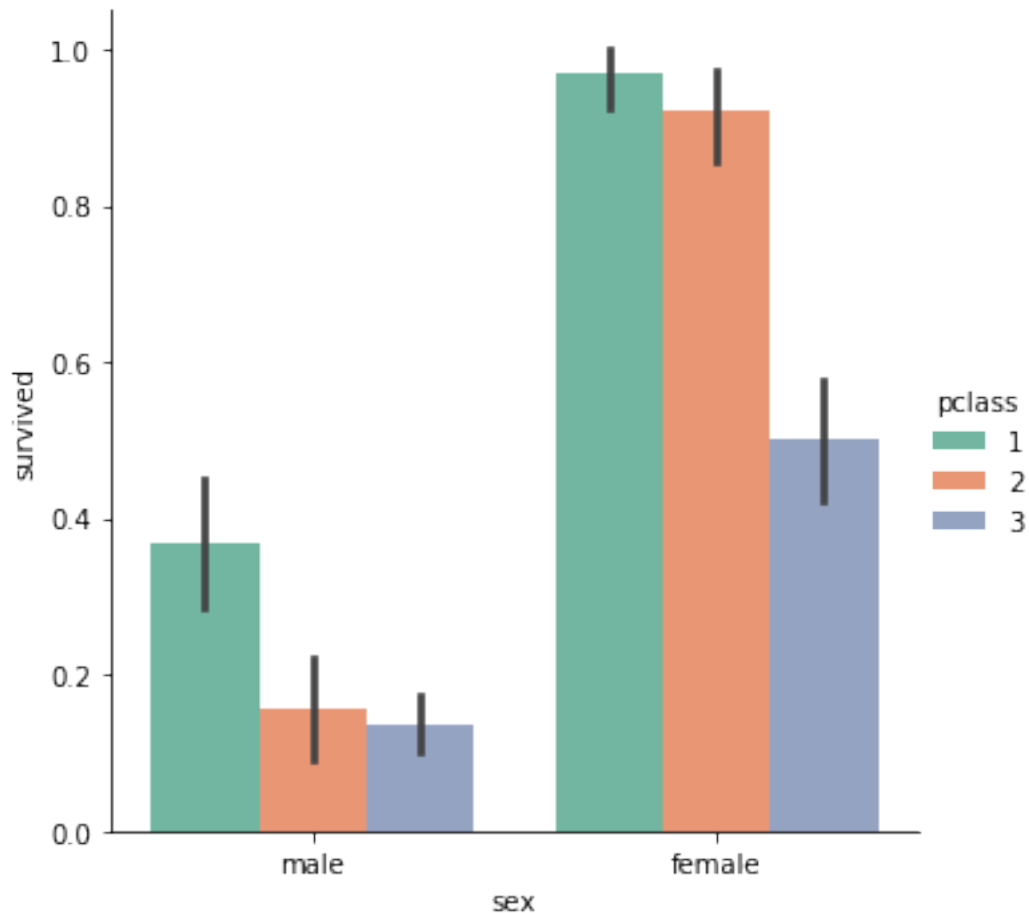
Ingresa el numero de la opcion: 4

Paleta Seleccionada

¡Vuelva pronto!

Probamos la paleta cualitativa con un dataset de los sobrevivientes del titanic, la cual es una grafica de barras que tiene en el eje x hombres y mujeres. Al no tener uno mas relevancia que otro, nos conviene ocupar una paleta cualitativa

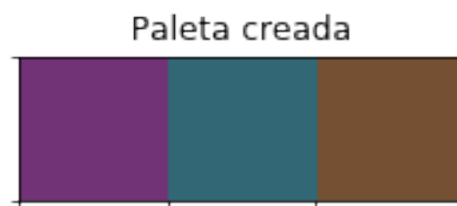
```
[50]: titanic = sns.load_dataset("titanic")
sns.catplot(x = "sex", y = "survived", kind = "bar", data = titanic, hue = "pclass", palette=sns.color_palette())
```



Ahora probamos la opción 2 de las paletas cualitativas, la cual es “Crear paleta”. Esta nos permite hacer nuestra propia paleta cualitativa, ingresando nuestros colores en formato hexadecimal separados por espacios.

[53]: `paleta_brewer.run()`

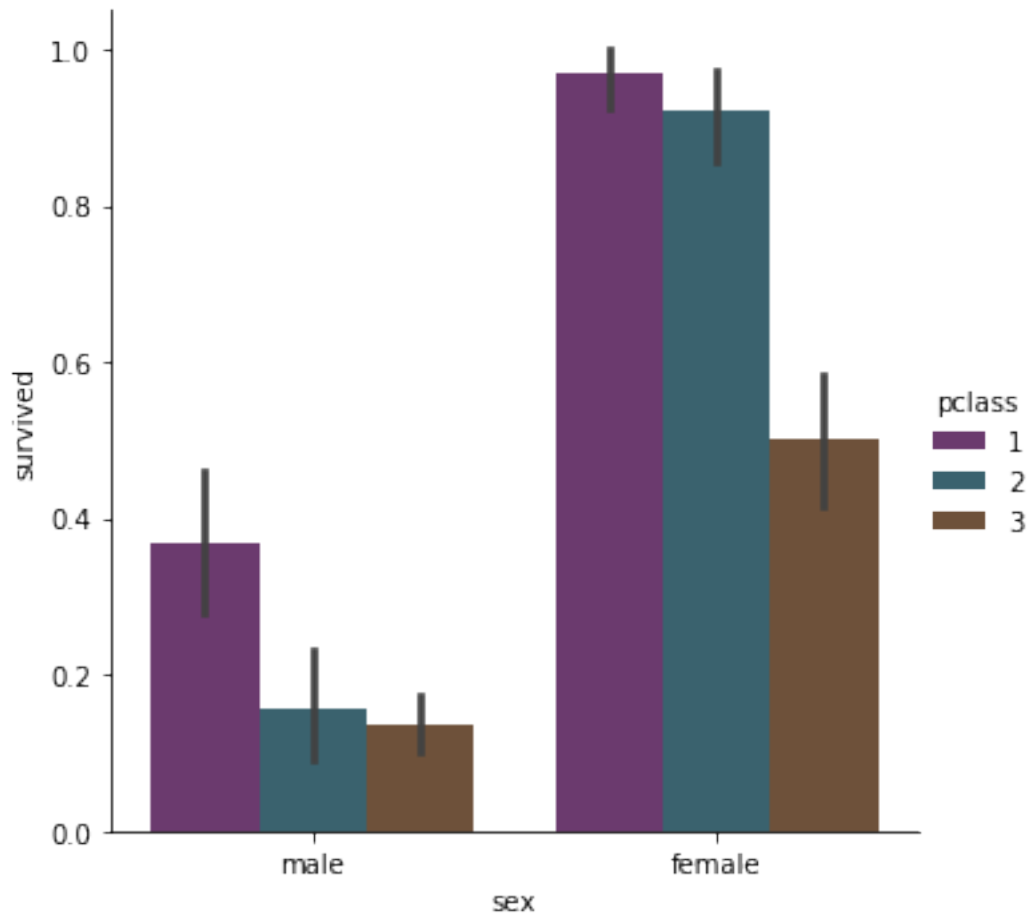
Ingrese los colores en formato hexadecimal, separados por espacios: #723276
 #326776 #765032
 Paleta Seleccionada
 ¿Vuelva pronto!



Probamos la paleta creada

```
[54]: titanic = sns.load_dataset("titanic")
sns.catplot(x = "sex", y = "survived", kind = "bar", data = titanic, hue = "pclass", palette=sns.color_palette())
```

```
[54]: <seaborn.axisgrid.FacetGrid at 0x7f86c24085d0>
```



0.1.4 Paleta especial para el daltonismo

Ingresamos la opción 2, llamada "Paleta colorblind friendly" la cual nos configura la paleta de colores especial para la deficiencia en la percepción de los colores

```
[56]: paleta_brewer.run()
```

Paleta Seleccionada
¡Vuelva pronto!



La probamos para con el mismo dataset titanic de arriba.

```
[57]: sns.catplot(x = "sex", y = "survived", kind = "bar", data = titanic, hue = "pclass", palette=sns.color_palette())
```

```
[57]: <seaborn.axisgrid.FacetGrid at 0x7f86c4947910>
```

