1 Visualizing all the Seaborn Color Palettes

In [1]:
```
from random import seed
from random import randint

import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("whitegrid")
%matplotlib inline

# Create a set of random numbers
seed(8)
i = 0
values= []
for i in range(100):
    value = randint(0, 10)
    values.append(value)
    i += 1

palette_options = ['Accent','Blues','BrBG','BuGn','BuPu','CMRmap','Dark2',
                   'GnBu','Greens','Greys','OrRd','Oranges','PRGn','Paired',
                   'Pastel1','Pastel2','PiYG','PuBu', 'PuBuGn','PuOr','PuRd',
                   'Purples','RdBu','RdGy','RdPu','RdYlBu','RdYlGn','Reds',
                   'Set1','Set2','Set3','Spectral','Wistia', 'YlGn', 'YlGnBu',
                   'YlOrBr','YlOrRd','afmhot','autumn','binary', 'bone','brg',
                   'bwr','cividis','cool','coolwarm','copper','crest',
                   'cubehelix','flag','flare','gist_earth','gist_gray',
                   'gist_heat','gist_ncar','gist_rainbow','gist_stern',
                   'gist_yarg','gnuplot', 'gnuplot2','gray','hot','hsv',
                   'icefire','inferno','magma','mako','nipy_spectral',
                   'ocean','pink','plasma','prism','rainbow','rocket',
                   'seismic','spring','summer','tab10', 'tab20', 'tab20b',
                   'tab20c','terrain','turbo', 'twilight','twilight_shifted',
                   'viridis','vlag','winter']

print ('Number of options: ', len(palette_options))
```

executed in 633ms, finished 15:47:27 2021-01-10

```
Number of options:  88
```

In [2]:
```
# Uncomment the code in this cell if you want to see all the palettes
# in only 1 column

# for option in palette_options:
#     sns.countplot(x= values, palette= option)
#     plt.title('Palette: {}'.format(option))
#     plt.show()
```

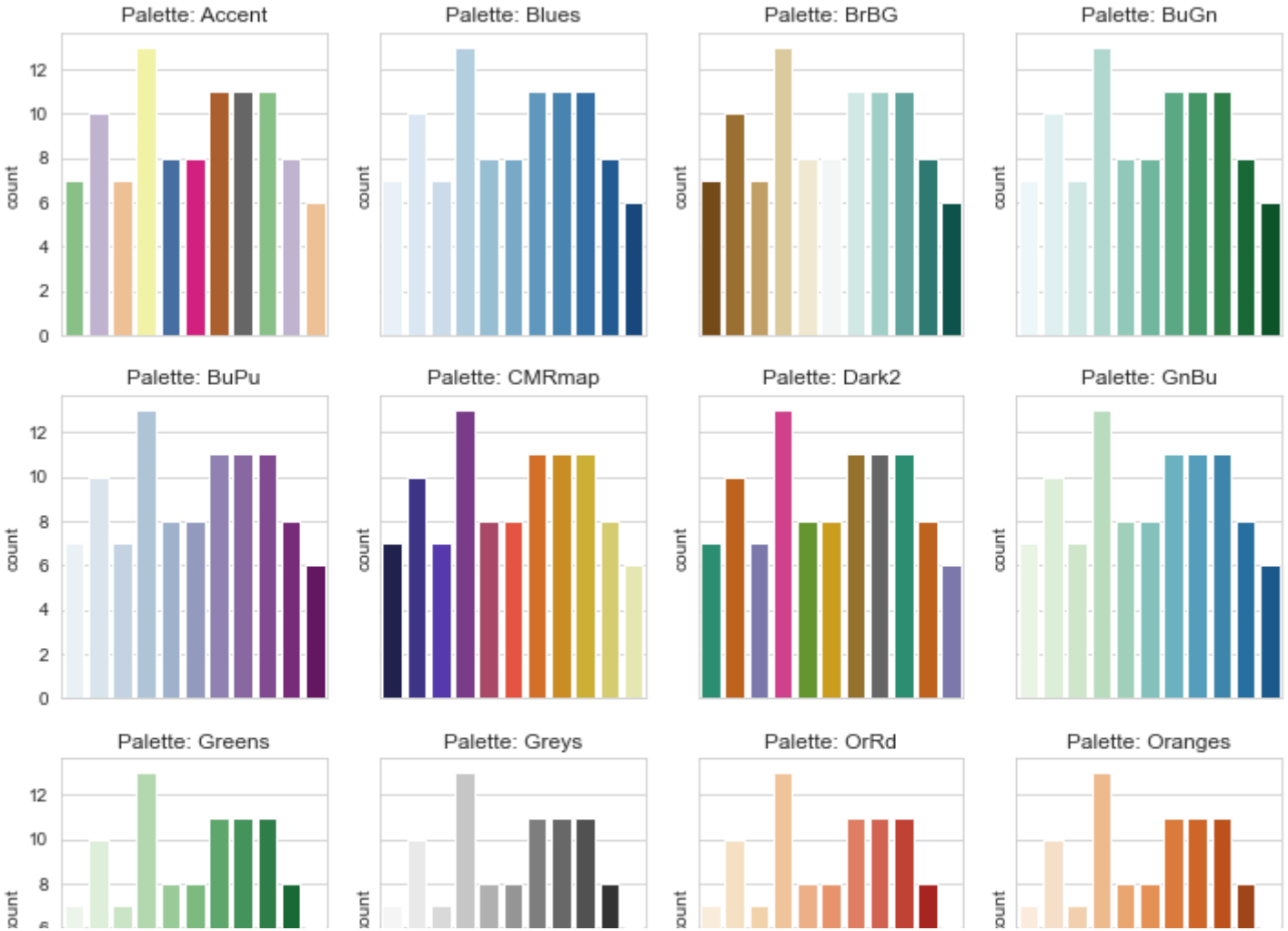
executed in 2ms, finished 15:47:27 2021-01-10

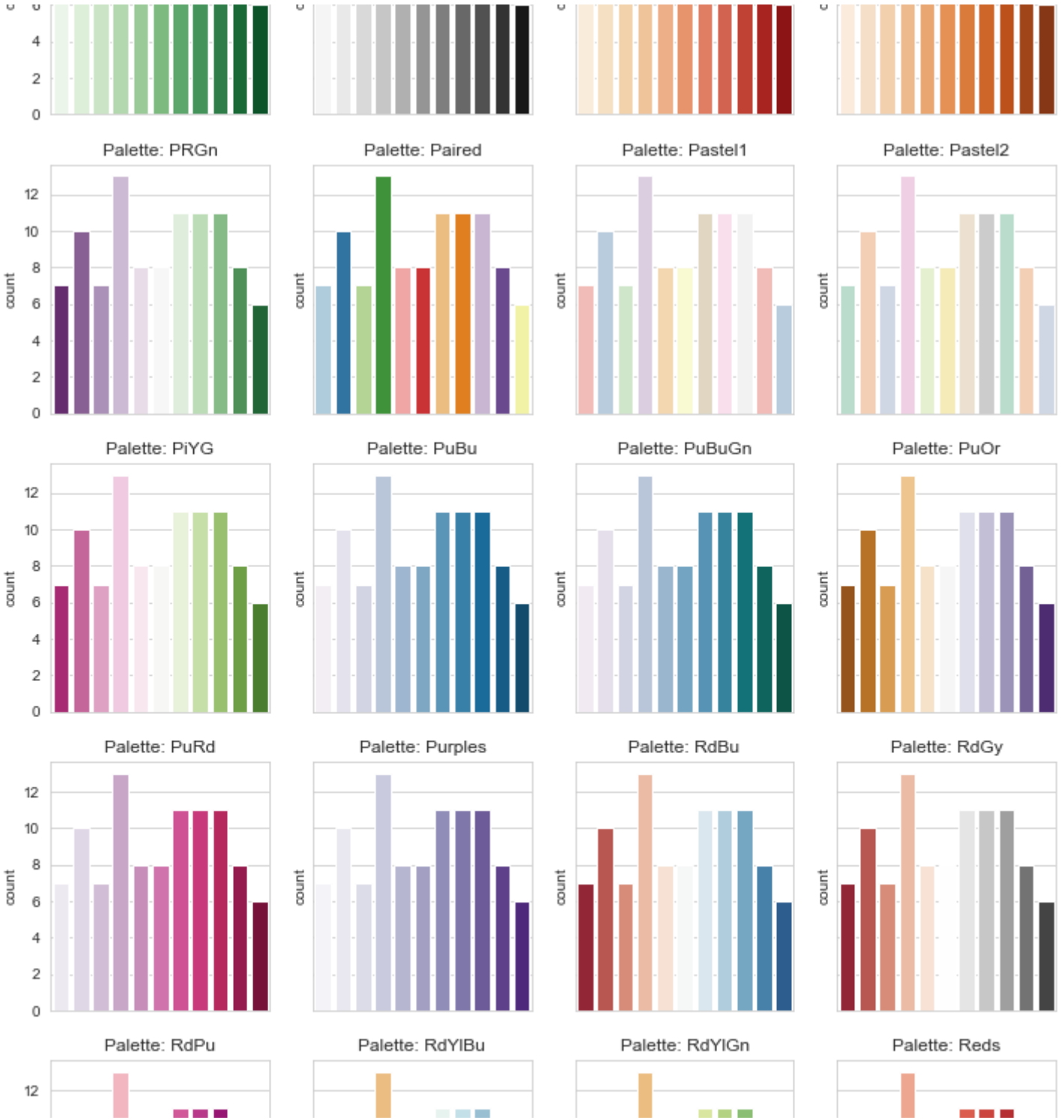
In [3]:
```
# Uncomment the code in this cell if you want to see all the palettes
# in 2 columns  (my favorite)

fig, axes = plt.subplots(nrows= 44, ncols= 2, figsize= (10, 200),
                         sharex= True, sharey= True)

# for ax, option in zip(axes.flatten(), palette_options):
#     sns.countplot(x= values, palette= option, ax= ax)
#     ax.set(title = 'Palette: {}'.format(option))
```

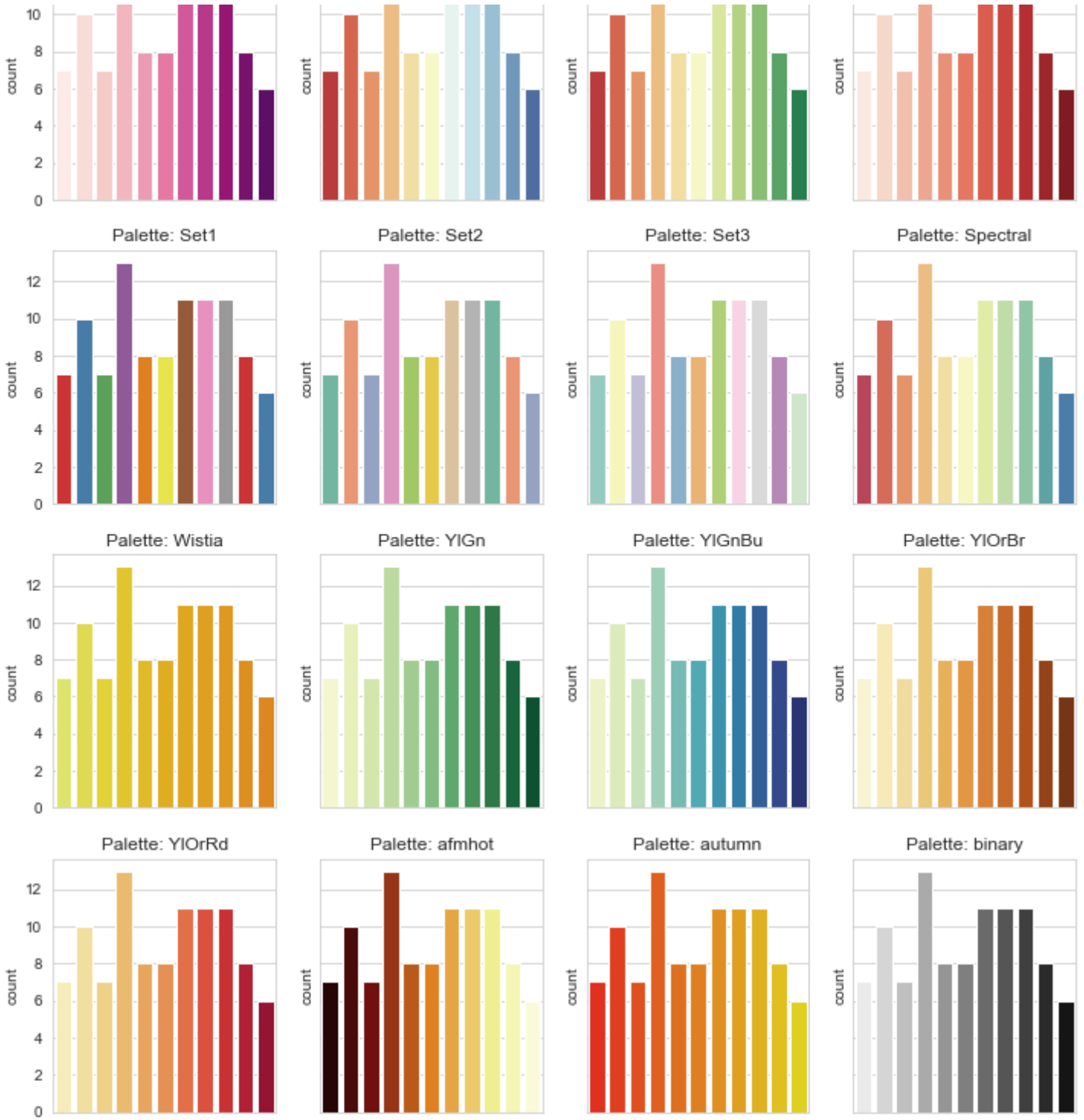
executed in 2ms, finished 15:47:27 2021-01-10

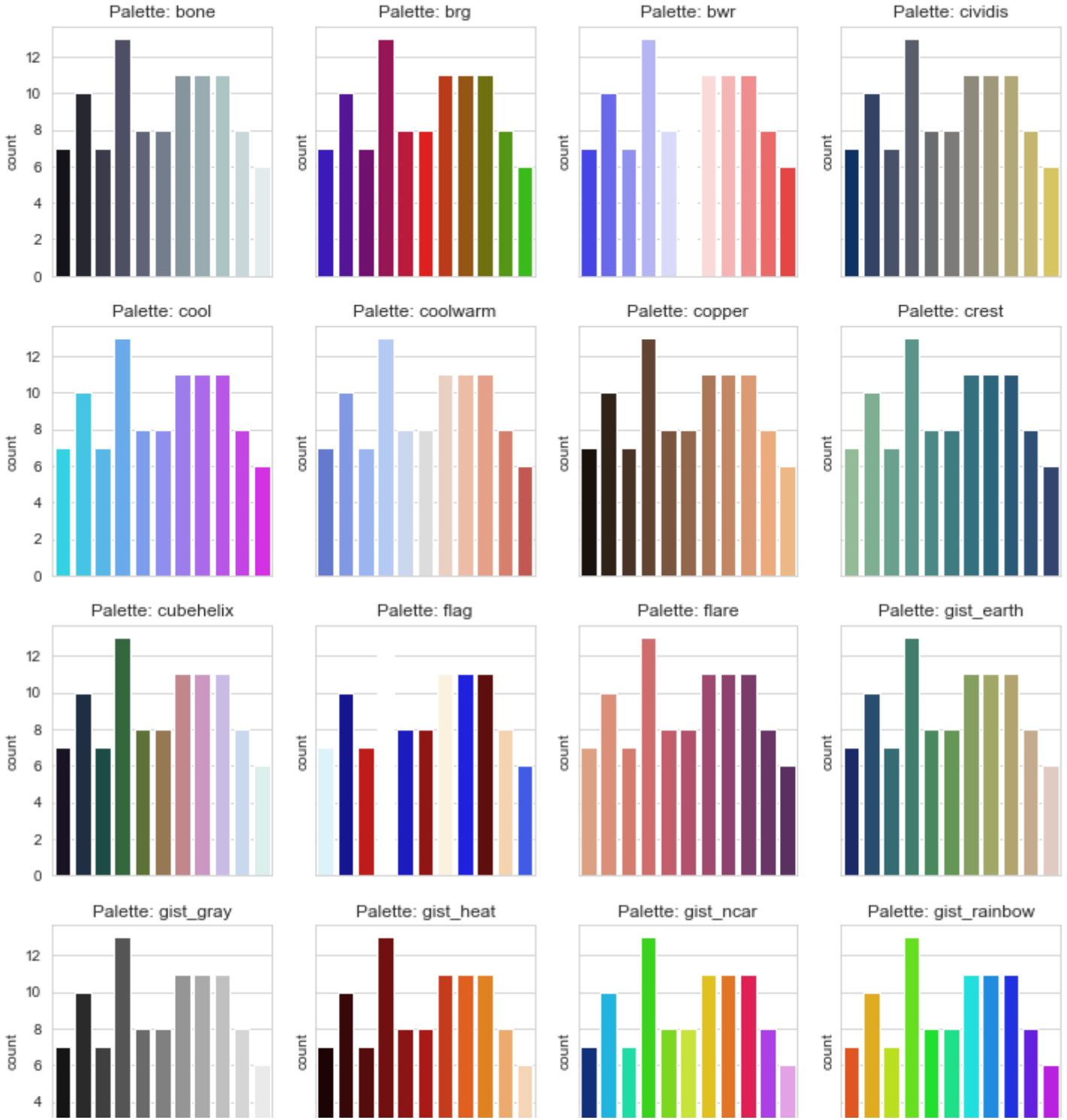
In [6]:
```
# Comment out this code if you are viewing palettes in only 1 or 2 columns

# This code will run as the default to reduce the amount of scrolling
# Please be patient - it takes a few seconds

fig, axes = plt.subplots(nrows= 22, ncols= 4, figsize= (12, 80), sharex= True,
                         sharey= True)

for ax, option in zip(axes.flatten(), palette_options):
    sns.countplot(x= values, palette= option, ax= ax)
    ax.set(title = 'Palette: {}'.format(option))
```

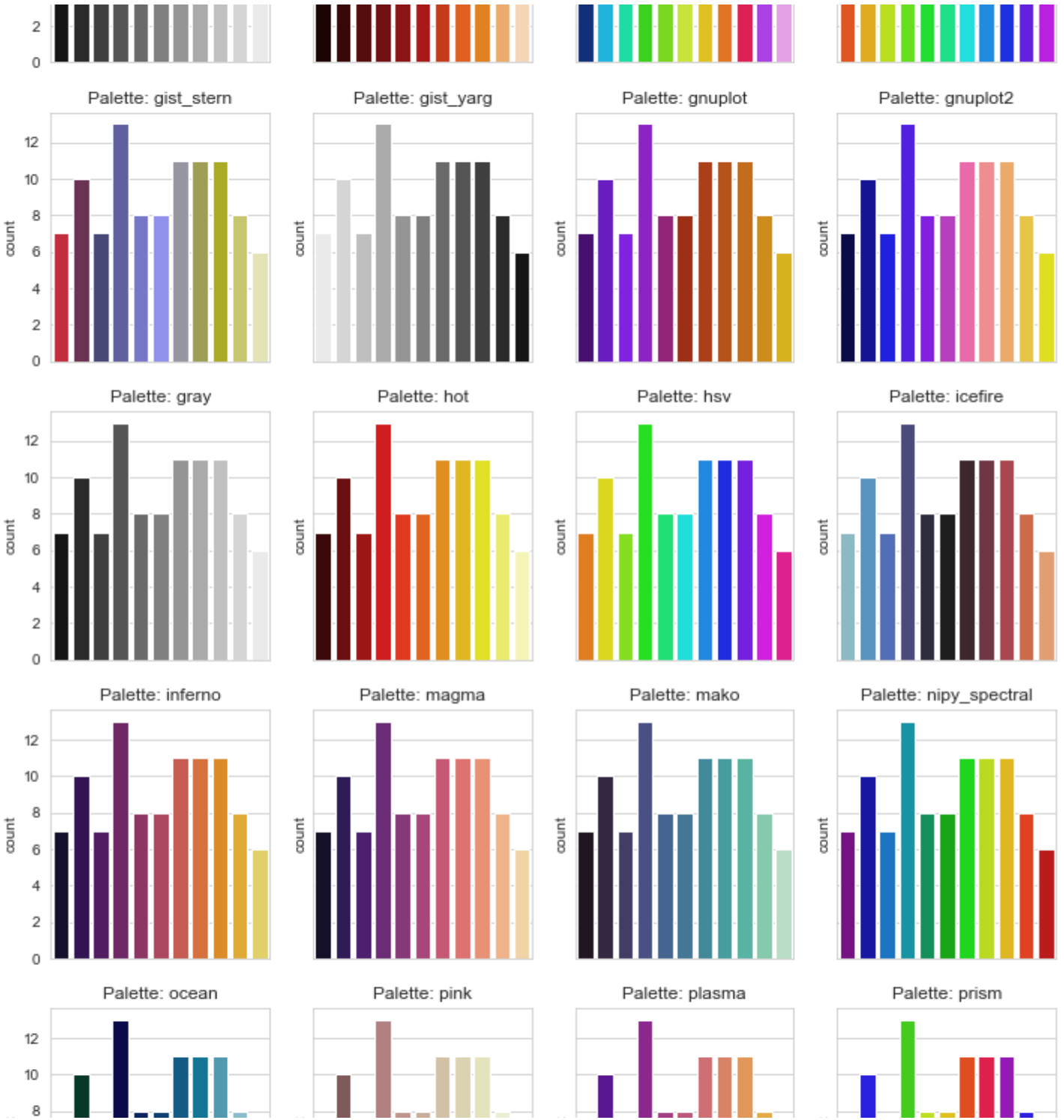
executed in 9.25s, finished 15:49:33 2021-01-10

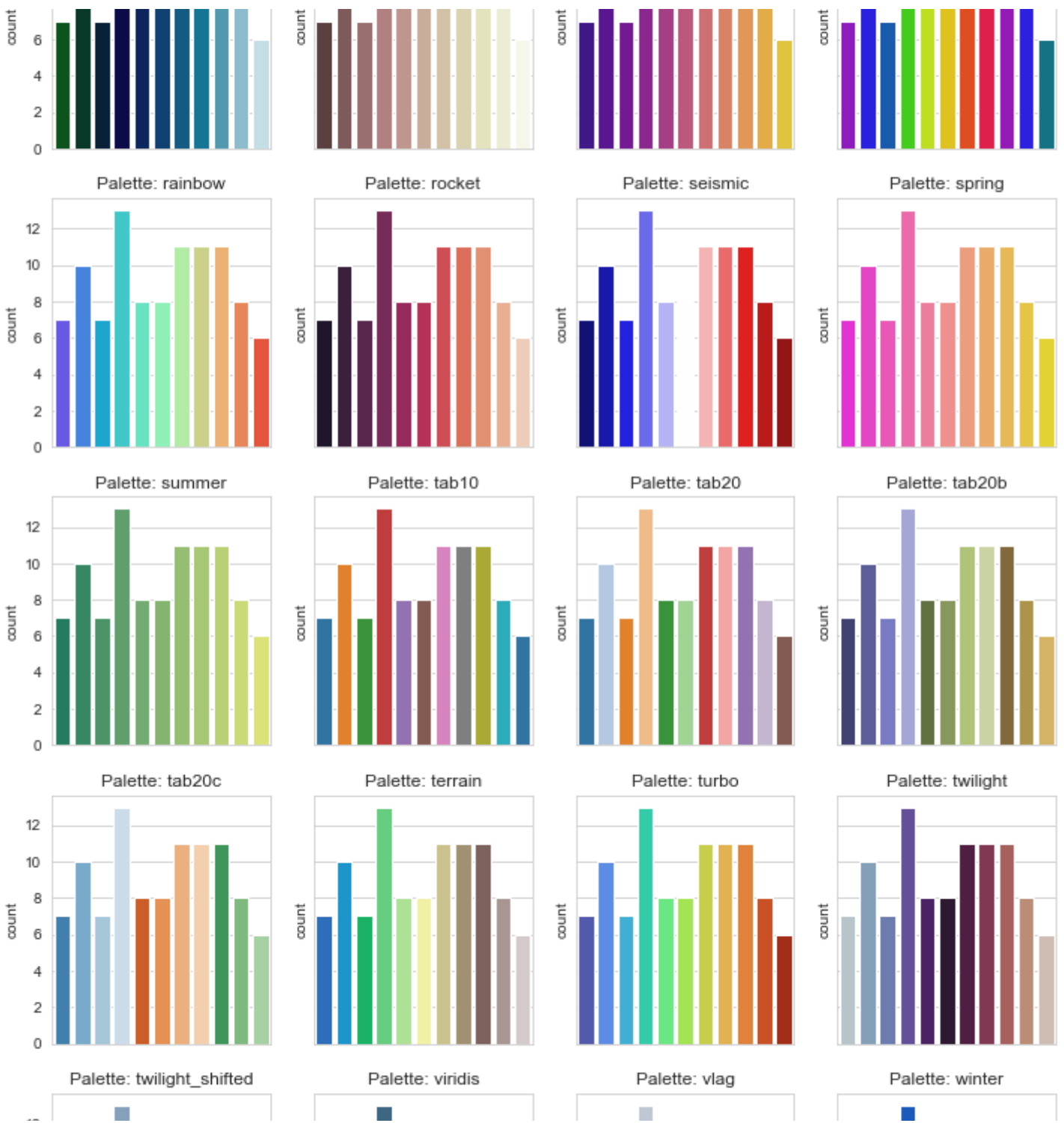


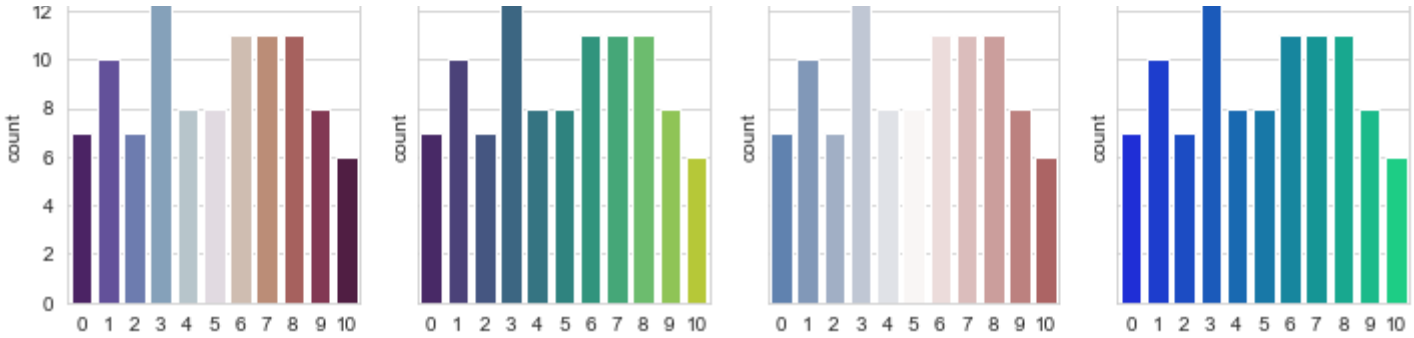

Palette: PRGn
Palette: Paired
Palette: Pastel1
Palette: Pastel2
Palette: PiYG
Palette: PuBu
Palette: PuBuGn
Palette: PuOr
Palette: PuRd
Palette: Purples
Palette: RdBu
Palette: RdGy
Palette: RdPu
Palette: RdYlBu
Palette: RdYlGn
Palette: Reds
count


Palette: Set1
Palette: Set2
Palette: Set3
Palette: Spectral
Palette: Wistia
Palette: YlGn
Palette: YlGnBu
Palette: YlOrBr
Palette: YlOrRd
Palette: afmhot
Palette: autumn
Palette: binary
count


Palette: bone
Palette: brg
Palette: bwr
Palette: cividis
Palette: cool
Palette: coolwarm
Palette: copper
Palette: crest
Palette: cubehelix
Palette: flag
Palette: flare
Palette: gist_earth
Palette: gist_gray
Palette: gist_heat
Palette: gist_ncar
Palette: gist_rainbow
count


Palette: gist_stern
Palette: gist_yarg
Palette: gnuplot
Palette: gnuplot2
Palette: gray
Palette: hot
Palette: hsv
Palette: icefire
Palette: inferno
Palette: magma
Palette: mako
Palette: nipy_spectral
Palette: ocean
Palette: pink
Palette: plasma
Palette: prism
count


Palette: rainbow
Palette: rocket
Palette: seismic
Palette: spring
Palette: summer
Palette: tab10
Palette: tab20
Palette: tab20b
Palette: tab20c
Palette: terrain
Palette: turbo
Palette: twilight
Palette: twilight_shifted
Palette: viridis
Palette: vlag
Palette: winter
count



2 Reverse the Colors using _r

Another option would be to reverse the order of the colors by adding ***_r*** to the end of any palette name.

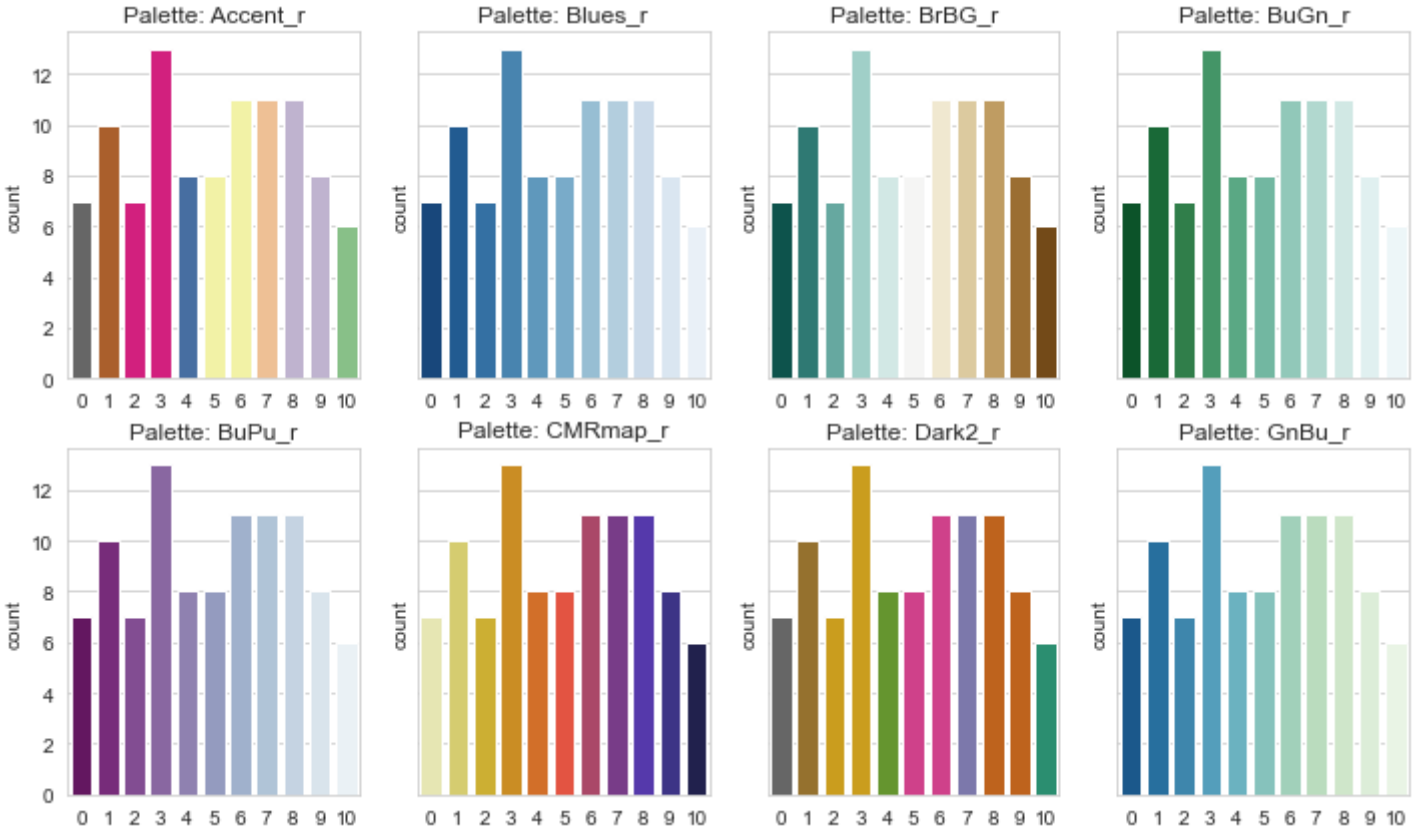
For example:

- palette "RdYlGn" goes from red to yellow to green
- palette "RdYlGn_r" goes from green to yellow to red

See the code below for a visual example.

In [5]:
```
# Demonstrates how adding _r to any palette will reverse the color's order
palette_options_reversed = ['Accent_r','Blues_r','BrBG_r','BuGn_r',
                            'BuPu_r','CMRmap_r','Dark2_r','GnBu_r']

fig, axes = plt.subplots(nrows= 2, ncols= 4, figsize= (12, 7), sharex= True,
                         sharey= True)

for ax, option in zip(axes.flatten(), palette_options_reversed):
    sns.countplot(x= values, palette= option, ax= ax)
    ax.set(title = 'Palette: {}'.format(option))
```

executed in 870ms, finished 15:47:38 2021-01-10



You can compare the original color order above to the order of the colors here and see they have switched.