

```
In [ ]: # Here we draw the directions graph with 8 point slices

N = len(summ8) # Length of our hypothetical dataset contained in a pandas series

# theta is used to get the x-axis values (the small part near the center of the chart)
# the np.linspace function is a nifty thing which evenly divides a number range by as many parts as you want
# in the example below we start at 0, stop at 2 * pi, and exclude the last number
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)

# width is used to get the widths of the top of the bars
# the value to divide by may need a little tinkering with, depending on the data
# so that the bars don't overlap each other
width = np.pi / 60 * summ8.values

# this quick update is to avoid low value bars being so skinny that you can't see them!
for i in range(len(width)):
    if width[i] < 0.2:
        width[i] = 0.2

plt.figure(figsize = (10, 10))
ax = plt.subplot(111, projection='polar')

# so here we specify theta for the x-axis, our series summ8 for the heights, the width is what we calculated above
bars = ax.bar(theta, summ8.values, width=width, bottom=0.0)

# this function makes sure that 0 degrees is set to North (the default is East for some strange reason)
ax.set_theta_zero_location("N")

# this function makes sure the degrees increase in a clockwise direction
ax.set_theta_direction(-1)

# and then these 2 options control how many radii there are and how they are labeled
# this may also need some tinkering depending on the data
ax.set_yticks(np.arange(0,20,3))

# and finally we do something simple and give it a title!
plt.title("Number of things by direction", fontsize = 15)

# Use custom colors and opacity
for r, bar in zip(summ8.values, bars):
    bar.set_facecolor(plt.cm.viridis(r / 13))
    bar.set_alpha(0.8)

plt.show()
```