Proficiency and frequency of use tabs data will be categorized into Complex bridge and Bridge Structures Groups.

CB BS

RETAIN THE TOTALS OF EACH ON A PER COLUMN BASIS

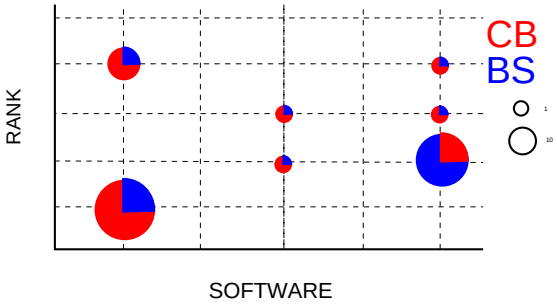
```
rawdf=readexcel.(path)
#use a combination of if statements and the pd.count() method to remove/ignore null values

def ccountingfunction():
    for i in each column matching each rant in cleandf:
        if i = CB:
            pd.count() # make a count of all the CBs
        if i = BS
            pd.count()
        else:
            print('errror')
    return a dataframe
cleandf= countingfunction(rawdf)
```

https://matplotlib.org/stable/gallery/lines_bars_and_markers/scatter_piecharts.html
once we have a dataframe that includes a count for each rank and software by group we can convert that to arcs.

Below looks like a simpler method

https://stackoverflow.com/questions/56337732/how-to-plot-scatter-pie-chart-using-matplotlib

```
import matplotlib.pyplot as plt

# first define the ratios
r1 = 0.2       # 20%
r2 = r1 + 0.4  # 40%

x = list(range(3))
y = list(range(3))

fig, ax = plt.subplots()

for xi,yi in zip(x,y):
    ax.pie([r1,r2,r2], colors=['indigo', "gold", 'crimson'],
           center=(xi, yi), radius=0.2+xi/4,
           wedgeprops=dict(width=(0.2+xi/4)/2), frame=True)
ax.autoscale()

plt.show()
```



WHAT IS A * IN THE MATPLTLIB DOCUMENTATION?

https://datascience.stackexchange.com/questions/28510/rectangular-markers-in-bubble-plot-python

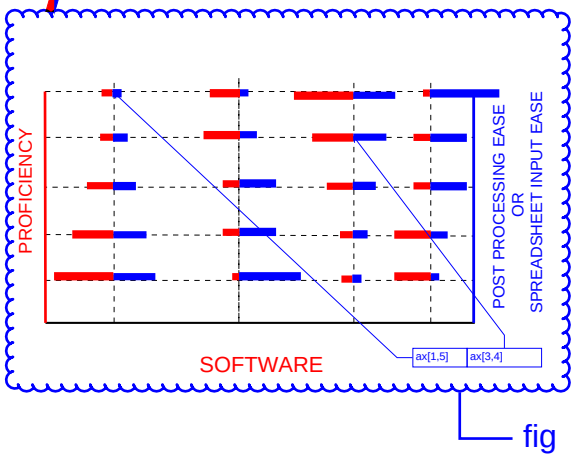
THIS IS RECCOMENDING NOT TO PLOT THE BARS AS MARKERS, BUT INSTEAD MAKE A FIGURE WITH EACH AX LOCATION CORRESPONDING TO A SOFTWARE AND PRFICIENCY

NON PROFICIENT USERS FINDING IT EASY TO DO POST PROCESSING AND SPREADSHEET INPUT

PEOPLE THINK THE SOFTWARE IS EASY TO WORK WITH UNTIL THEY BECOME PROFICIENT

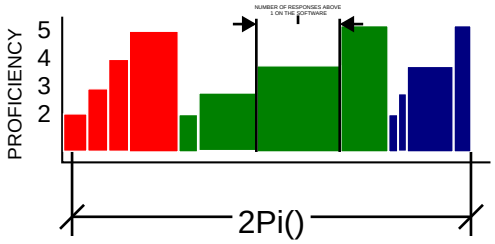
THOSE THAT ARE PROFICIENT FINDING IT EASY TO DO POST PROCESSING AND SPREADSHEET INPUT

PEOPLE WHO ARE PROFICIENT WITH THE SOFTWARE FIND IT EASY TO WORK WITH



RANK SOFTWARES BY THE NUMBER OF RESPONSES ABOVE ONE SO THAT THE PIE CHART WILL COMPARE THINGS OF SIMILAR POPULARITY NEAR ONE ANOTHER

ax=plt.subplot(projection=polar)
ax.bar(x,height=rank,width=number of responses, align left, spacebetween=0)



ONCE THE BAR CHART IS MADE IN THE CARTISIAN PLANE THE PROJECTION TO POLAR WILL MAKE IT LOOK AND ACT LINE A PIE CHART