In [2]:
```
'''A pie chart is used for plotting numerical proportions.
we use pie() function in python to draw a pie chart'''
#step1: importing  libraries
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
#step2: loading excel data set in jupyter environment
dataset = pd.read_excel("C:/Users/subed/OneDrive/Desktop/garments_worker_productiv
```

In [4]:
```
#step3: calling dataset
dataset
```

Out[4]:

	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time
0	2015-01-01	Quarter1	sweing	Thursday	8	0.80	26.16	1108.0	7080
1	2015-01-01	Quarter1	finishing	Thursday	1	0.75	3.94	NaN	960
2	2015-01-01	Quarter1	sweing	Thursday	11	0.80	11.41	968.0	3660
3	2015-01-01	Quarter1	sweing	Thursday	12	0.80	11.41	968.0	3660
4	2015-01-01	Quarter1	sweing	Thursday	6	0.80	25.90	1170.0	1920
...	...	...	...	...	...	...	...	...	...
1192	2015-03-11	Quarter2	finishing	Wednesday	10	0.75	2.90	NaN	960
1193	2015-03-11	Quarter2	finishing	Wednesday	8	0.70	3.90	NaN	960
1194	2015-03-11	Quarter2	finishing	Wednesday	7	0.65	3.90	NaN	960
1195	2015-03-11	Quarter2	finishing	Wednesday	9	0.75	2.90	NaN	1800
1196	2015-03-11	Quarter2	finishing	Wednesday	6	0.70	2.90	NaN	720

1197 rows × 15 columns

In [5]:
```
#display all the columns names in console
dataset.columns
#smv = standard minute value
#wip= work in progress
# over_time = amount of overtime done by each team in minute
# targeted_productivity: is productivity set by authority for each team for each d
# actual_productivity : is the productivity delivered by workers in % ( from 0 to
```

Out[5]:
```
Index(['date', 'quarter', 'department', 'day', 'team', 'targeted_productivity',
       'smv', 'wip', 'over_time', 'incentive', 'idle_time', 'idle_men',
       'no_of_style_change', 'no_of_workers', 'actual_productivity'],
      dtype='object')
```

```
In [10]: dataset.shape

Out[10]: (1197, 15)
```

```
In [11]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   date                   1197 non-null   datetime64[ns]
 1   quarter                1197 non-null   object
 2   department             1197 non-null   object
 3   day                    1197 non-null   object
 4   team                   1197 non-null   int64
 5   targeted_productivity  1197 non-null   float64
 6   smv                    1197 non-null   float64
 7   wip                    691 non-null    float64
 8   over_time              1197 non-null   int64
 9   incentive              1197 non-null   int64
 10  idle_time              1197 non-null   float64
 11  idle_men               1197 non-null   int64
 12  no_of_style_change     1197 non-null   int64
 13  no_of_workers          1197 non-null   float64
 14  actual_productivity    1197 non-null   float64
dtypes: datetime64[ns](1), float64(6), int64(5), object(3)
memory usage: 140.4+ KB
```

```
In [12]: #step:5 dataset cleaning
         dataset['quarter'].value_counts()

Out[12]: Quarter1    360
         Quarter2    335
         Quarter4    248
         Quarter3    210
         Quarter5     44
         Name: quarter, dtype: int64
```

```
In [13]: dataset['department'].value_counts().index.to_list() # each department

Out[13]: ['sweing', 'finishing ', 'finishing']
```

```
In [15]: # data cleaning(redundancies removing)
         dataset['department'] = dataset['department'].apply(lambda x: 'finishing' if x ==
         dataset['department'].value_counts().index.to_list()

Out[15]: ['sewing', 'finishing']
```

In [16]:

```
#Total record value count in each day
dataset['day'].value_counts()
```

Out[16]:

```
Wednesday    208
Sunday       203
Tuesday      201
Monday       199
Thursday     199
Saturday     187
Name: day, dtype: int64
```

In [17]:

```
#Total record value count of each department
dept = dataset.department.value_counts().reset_index()
dept.rename(columns = {'index':'department', 'department':'total_num'},inplace=Tru
dept
```

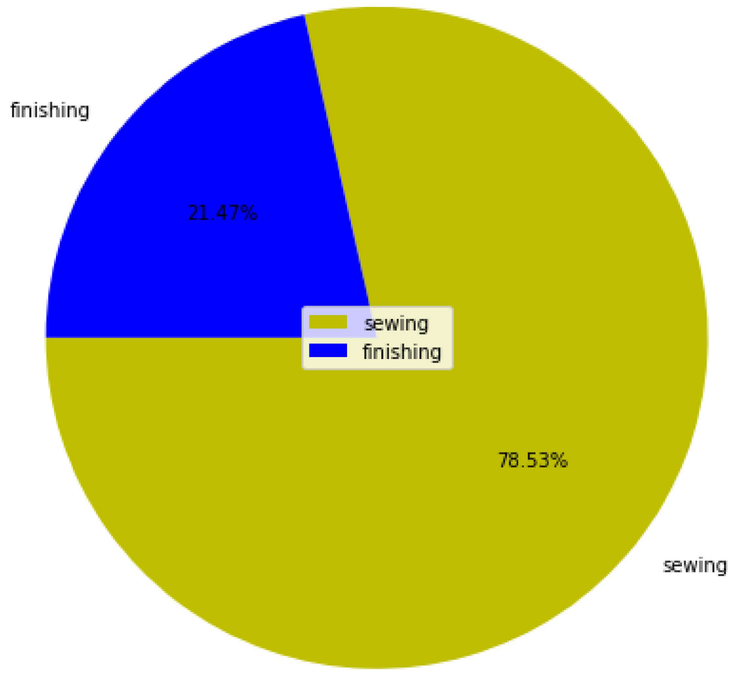
Out[17]:

	department	total_num
0	sewing	940
1	finishing	257

In [54]:
```
#step5: creting pieplot
# work1: This is the pie plot of department with total_nums of workers

activities = ['sewing','finishing'] # defining labels
slices= [940,257] # portion covered by each label
colors=['y','b'] # color of the slices
#plotting piechart
plt.pie(slices,labels= activities, colors = colors, explode = (0,0), startangle=18
        radius = 2, autopct = '%2.2f%%' )
# plotting legend
plt.legend(loc='center')
plt.show() # showing the plot

'''The pie chart result shows according to the displayed data, with colors, percen
This gives the quick visualization of the data output.'''
```



Out[54]:
```
'The pie chart result shows according to the displayed data, with colors, percen
tage and name\nThis gives the quick visualization of the data output.'
```

In [29]:
```
# total value count for each quarter
dataset['quarter'].value_counts()
```

Out[29]:
```
Quarter1    360
Quarter2    335
Quarter4    248
Quarter3    210
Quarter5     44
Name: quarter, dtype: int64
```

In [31]:

```
#dataset of quarter and value count made to create pie plot.
quart = dataset.quarter.value_counts().reset_index()
quart.rename(columns = {'index':'quarter', 'quarter':'days_in_each_quarter'},inpla
quart
```

Out[31]:

	quarter	days_in_each_quarter
0	Quarter1	360
1	Quarter2	335
2	Quarter4	248
3	Quarter3	210
4	Quarter5	44

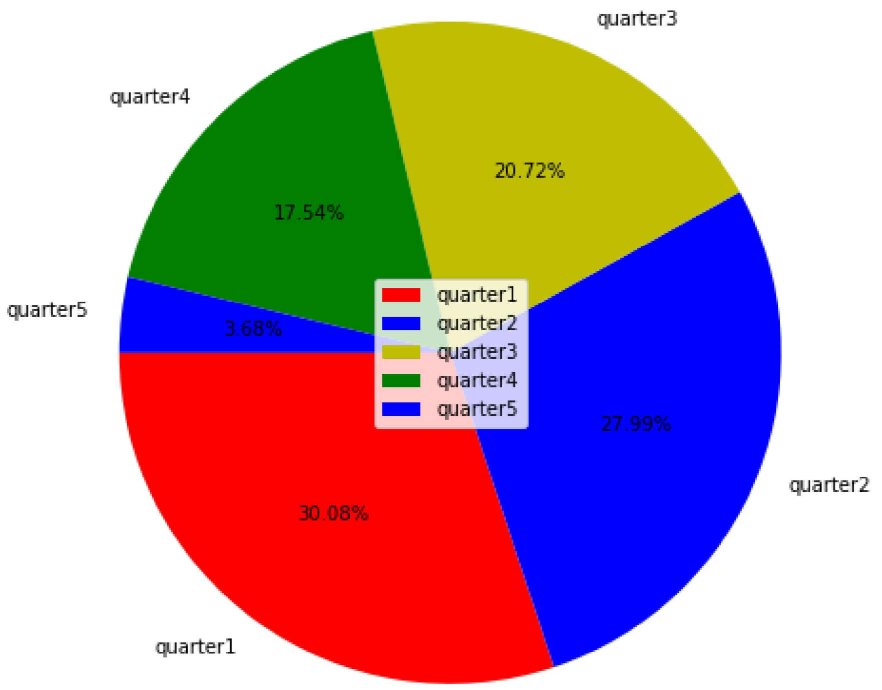
In [53]:

```
#step5: creting pieplot
# work2: This is the pie plot of department with total_nums of workers
activities = ['quarter1','quarter2','quarter3','quarter4','quarter5'] # defining l
slices= [360,335,248,210,44] # portion covered by each label
colors=['r','b','y','g','b'] # color of the slices

#plotting piechart
plt.pie(slices,labels= activities, colors = colors, explode = (0,0,0,0,0), startar
        radius = 2, autopct = '%2.2f%%' )
# plotting legend
plt.legend(loc='center')

plt.show() # showing the plot

#The pie chart result shows according to the displayed data, with colors, percenta
#this gives the quick visualization of the data output.
```



```
In [ ]:
```