Name-Arpit Aditya
Class-Data Analytics
Reg No-21BCE8700

Basic Setup

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data=pd.read_csv('21BCE8700 - Form Responses 1.csv')
print('The number of rows in the dataset is:')
print(data.index.size)
data=data.drop(['Score'],axis=1)
print('The number of columns in the dataset is:')
print(data.columns.size)#Printing the number of columns
data=pd.DataFrame(data)
print(data.isna().sum())#Printing the number of null values in each column
data['How do you think A.I will impact the economy?'].fillna("Improve Economy",inplace=True)
# print('after replacing na values')
#Dropping the first two columns
data=data.drop(['Timestamp'],axis=1)
# print(data.describe)
column_name=data.columns#Getting all the column names
print(column_name)
```

Output-

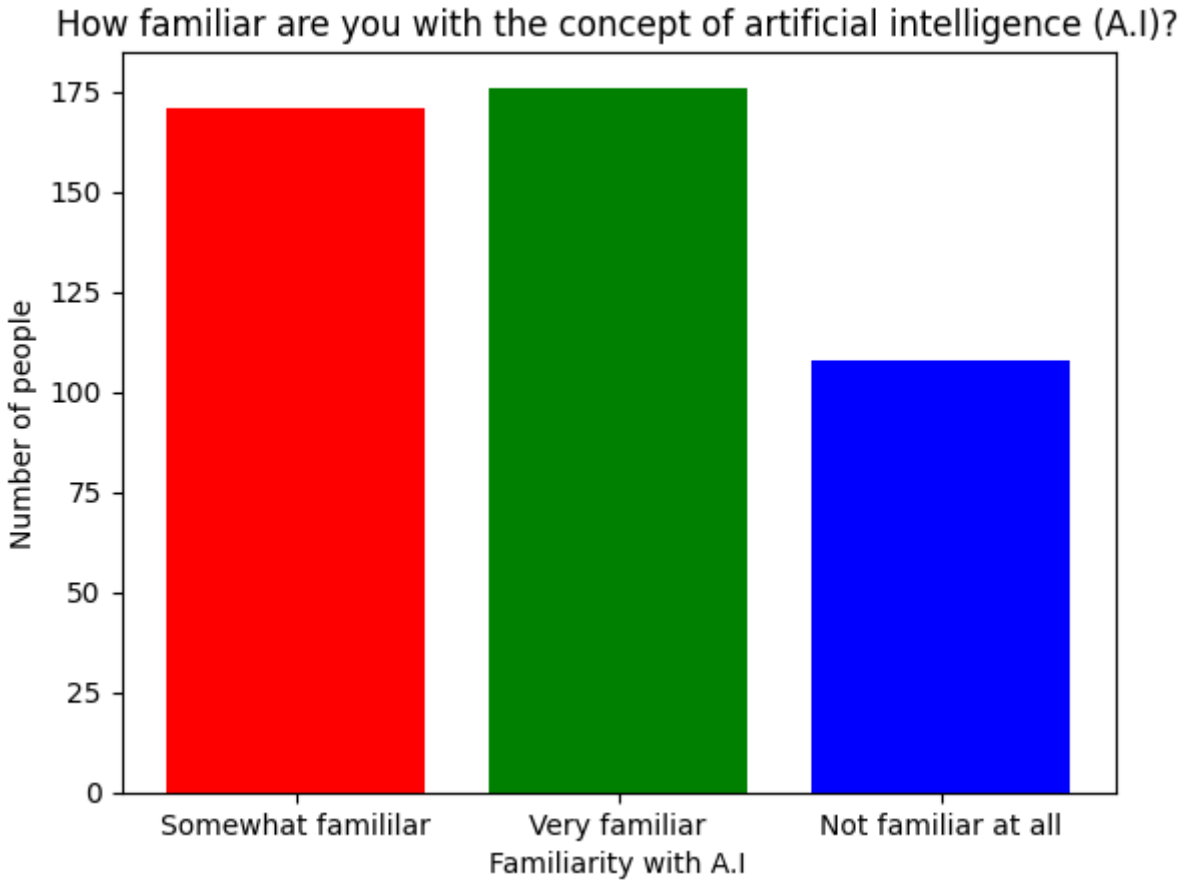
```
Timestamp                                                                                0
Email Address                                                                            0
How familiar are you with the concept of artificial intelligence (A.I)?                  0
In your opinion, will A.I have a positive impact on our future?                          0
How do you think A.I will affect job opportunities in the future?                        0
Do you think A.I will change the way we live our daily lives?                            0
In what industries do you think A.I will have the most impact?                           0
How do you think A.I will affect the way we interact with each other?                    0
Do you think A.I will be able to replicate human emotions and consciousness?             0
How do you think A.I will impact the way we make decisions?                              0
Do you think A.I will be able to make ethical decisions?                                 0
In your opinion, should there be regulations in place to govern the development and use of A.I?      0
How do you think A.I will impact privacy?                                                0
Do you think A.I will be able to replace human creativity?                               0
How concerned are you about the potential for A.I to be used for unethical or malicious purposes?    0
In your opinion, could A.I potentially become a threat to humanity?                      0
How do you think A.I will impact the economy?                                            2
dtype: int64
Index(['Email Address',
       'How familiar are you with the concept of artificial intelligence (A.I)?',
       'In your opinion, will A.I have a positive impact on our future?',
       'How do you think A.I will affect job opportunities in the future?',
       'Do you think A.I will change the way we live our daily lives?',
       'In what industries do you think A.I will have the most impact?',
       'How do you think A.I will affect the way we interact with each other?',
...
       'How concerned are you about the potential for A.I to be used for unethical or malicious purposes?',
       'In your opinion, could A.I potentially become a threat to humanity?',
       'How do you think A.I will impact the economy?'],
      dtype='object')
```

Q1-How many people are familiar with A.I?

Code-

```
#How many people are familiar with A.I?
data.iloc[:,1]
values=data.iloc[:,1].unique()
[a,b,c]=[data.iloc[:,1].value_counts()
[values[0]],data.iloc[:,1].value_counts()
[values[1]],data.iloc[:,1].value_counts()[values[2]]]
output=[a,b,c]#Creating Array of values
plt.bar(values,output,color=['red','green','blue'])
plt.ylabel('Number of people')
plt.xlabel('Familiarity with A.I')
plt.title(column_name[1])
plt.show()
```

Output-

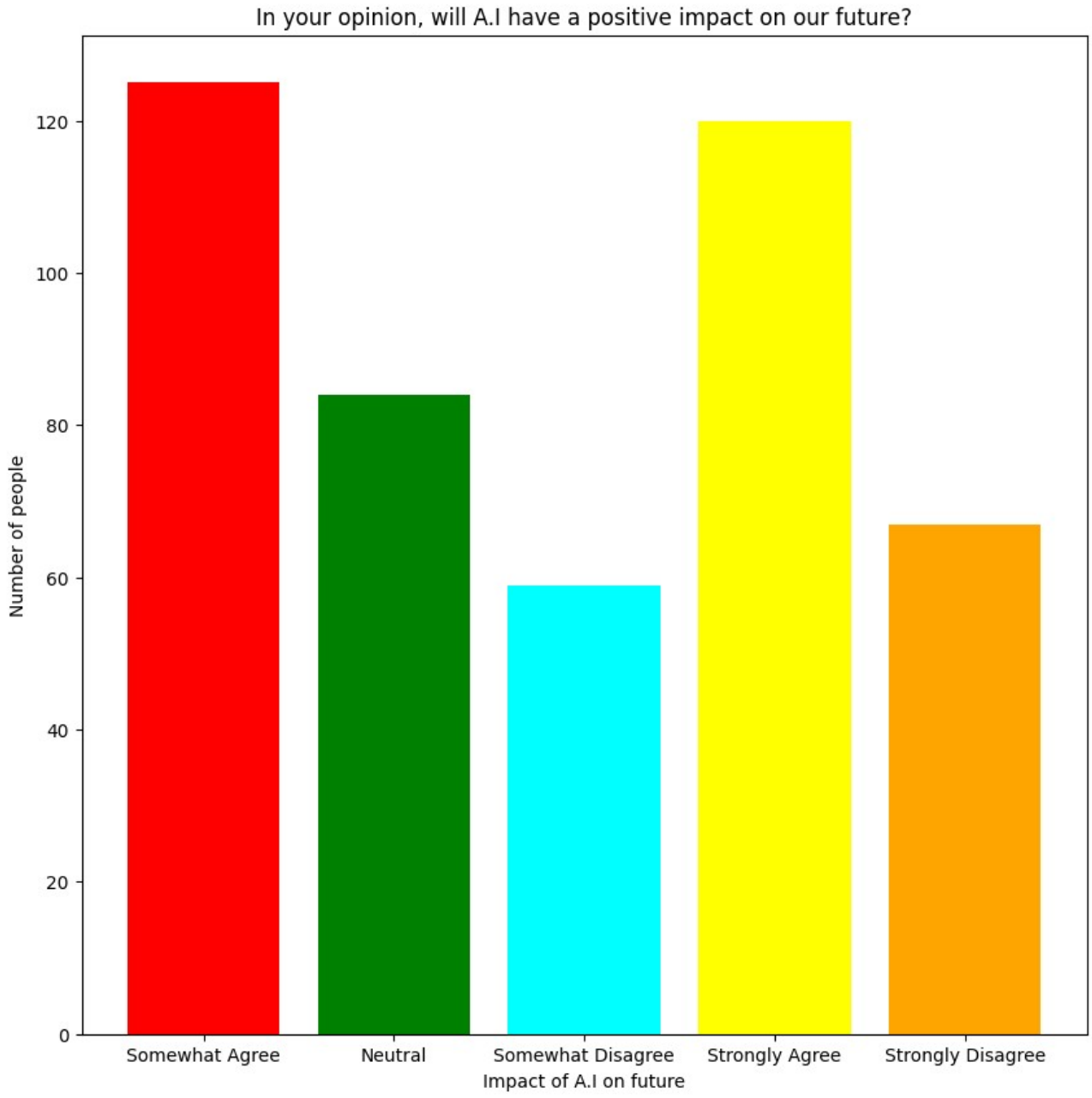


Q2-How many people think that A.I will have positive impact on future?

Code-

```
# Q2-How many people think that A.I will have positive impact on future?
data.iloc[:,2]
values=data.iloc[:,2].unique()
print(values)
output=[]
for i in values:
    temp=data.iloc[:,2].value_counts()[i]
    output.append(temp)

#Creating Array of values
plt.figure(figsize=(10, 10))
plt.bar(values,output,color=['red','green','cyan','yellow','orange'])
plt.ylabel('Number of people')
plt.xlabel('Impact of A.I on future')
plt.title(column_name[2])
plt.show()
```

Output-


In your opinion, will A.I have a positive impact on our future?
Number of people
0
20
40
60
80
100
120
Somewhat Agree
Neutral
Somewhat Disagree
Strongly Agree
Strongly Disagree
Impact of A.I on future

l

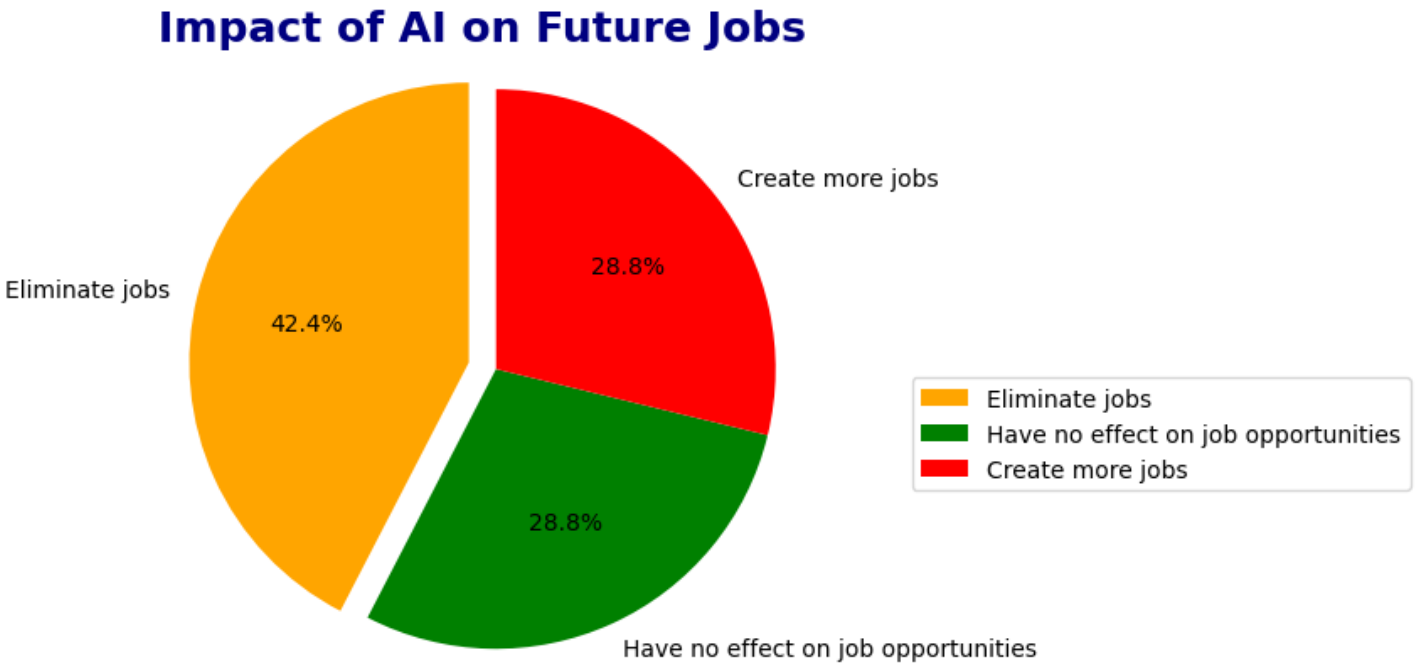
Q3-How many people think that A.I will have an positive impact on future and eliminate jobs?

Code-

```
#Q3-How many people think that A.I will have an positive impact on future and eliminate jobs?
data.iloc[:,2]#Counting all the values in the column
values=data.iloc[:,2].unique()#Getting the unique values
values_col_3=data.iloc[:,3].unique()#Getting the unique values of column 3
print(values_col_3)
positive_impact = data[data[column_name[2]] == values[0]]#Getting the data where the value is positive
eliminate_jobs=positive_impact[positive_impact[column_name[3]]==values_col_3[0]]#Getting the data where the value is eliminate jobs
print("The number of people who think that A.I will have a positive impact on future and eliminate jobs are:")
print(eliminate_jobs.index.size)
No_effect_jobs=positive_impact[positive_impact[column_name[3]]==values_col_3[1]]#Getting the data where the value is No effect on jobs
Create_more_jobs=positive_impact[positive_impact[column_name[3]]==values_col_3[2]]#Getting the data where the value is Create more jobs
sizes = [eliminate_jobs.index.size,No_effect_jobs.index.size,Create_more_jobs.index.size]#Storing all the responses in an array
labels=[values_col_3[0],values_col_3[1],values_col_3[2]]#Storing all the labels in an array
explode=(0.1,0,0)#Creating an array for explode
colors = ['#FFA500', '#008000', '#FF0000']#Storing all the colors in an array
```

```
plt.pie(sizes, labels=labels,
colors=colors,explode=explode,autopct='%1.1f%%',
startangle=90)
plt.title('Impact of AI on Future Jobs', fontsize=18,
fontweight='bold', color='navy')
plt.legend(loc='best', bbox_to_anchor=(1, 0.5))
plt.axis('equal')
plt.show()
```

Output-



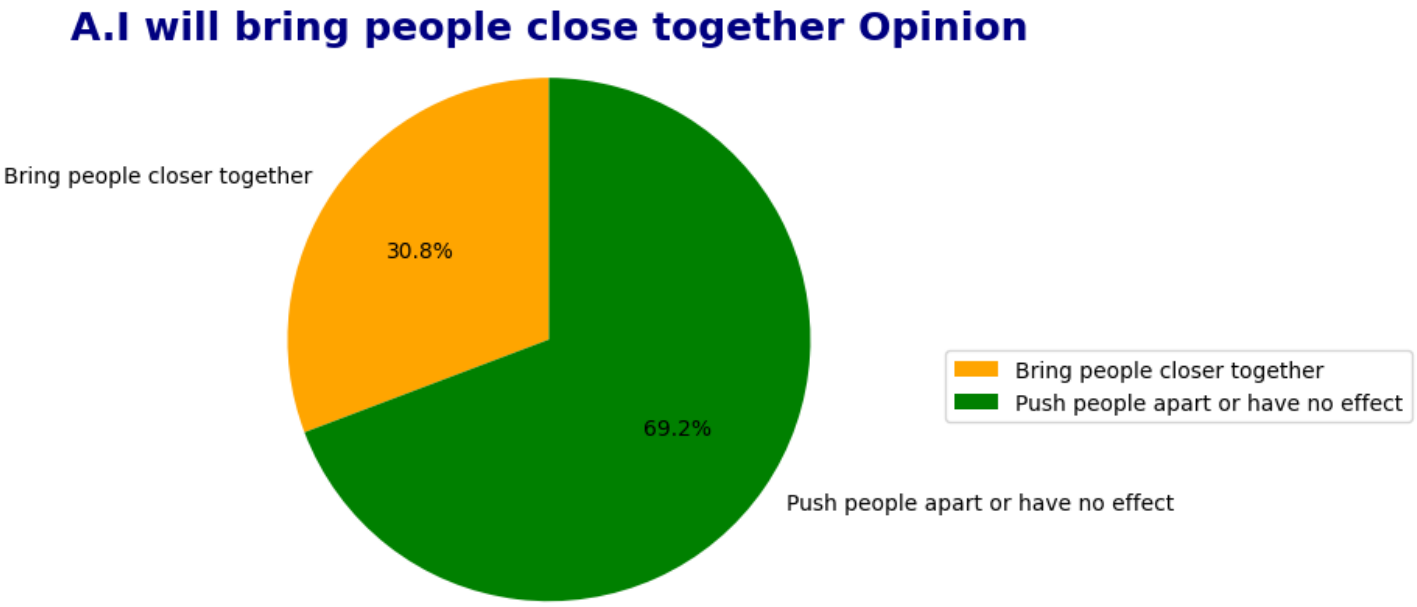
l

Q4-How many people are there who think that A.I will bring people close together?

Code-

```
# Q4-How many people are there who think that A.I will bring people close together?
print(column_name[6])
values_col_6=data.iloc[:,6].unique()#Getting the unique values of column 6
print(values_col_6)
Close_together=data[data[column_name[6]]==values_col_6[1]]#Getting the data where the value is Bring people close together
print("The number of people who think A.I will bring people close together are:")
print(Close_together.index.size)
Remaining_people=data.index.size-Close_together.index.size#Getting the remaining people
sizes=[Close_together.index.size,Remaining_people]#Storing all the responses in an array
colors=['#FFA500','#008000']#Storing all the colors in an array
labesl=[values_col_6[1],'Push people apart or have no effect']#Storing all the labels in an array
plt.pie(sizes,labels=labesl,colors=colors,autopct='%1.1f%%',startangle=90)
plt.title('A.I will bring people close together Opinion', fontsize=18, fontweight='bold', color='navy')
plt.legend(loc='best', bbox_to_anchor=(1, 0.5))
plt.axis('equal')
plt.show()
```

Output-

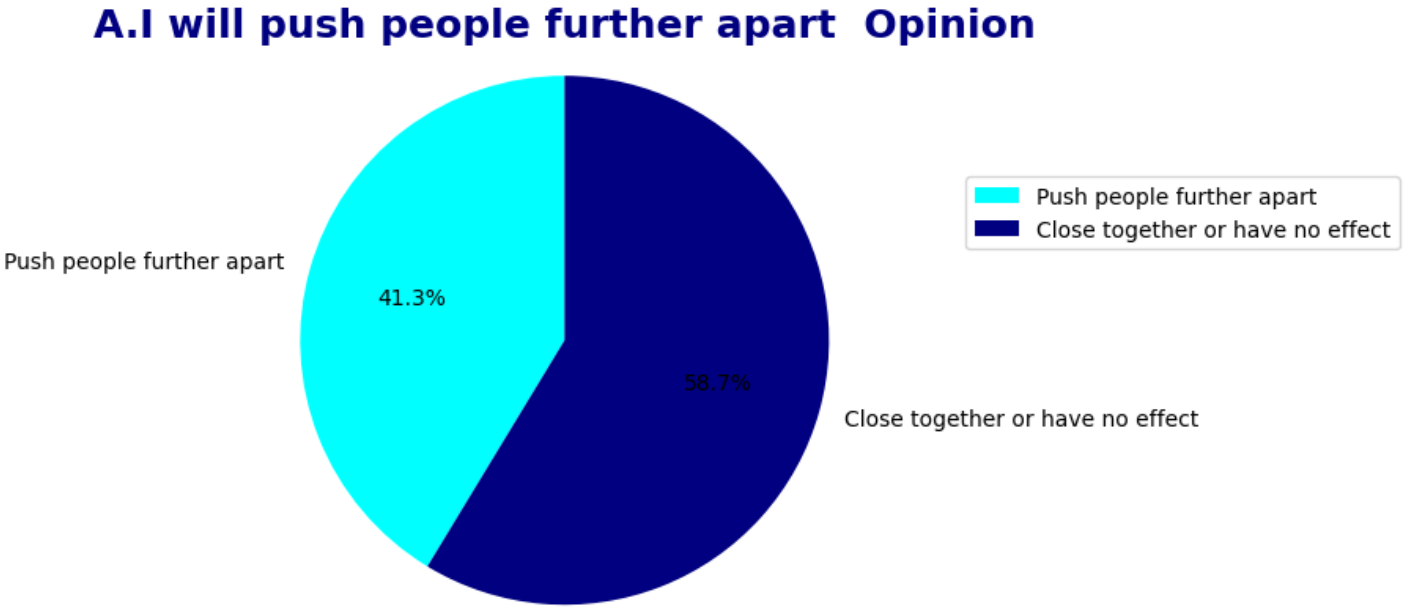

A.I will bring people close together Opinion
Bring people closer together
30.8%
69.2%
Push people apart or have no effect
Bring people closer together
Push people apart or have no effect

l

Q5-How many people are there who think that A.I will push people apart.

```
# Q5-How many people are there who think that A.I will Push people apart?
print(column_name[6])
values_col_6=data.iloc[:,6].unique()#Getting the unique values of column 6
print(values_col_6)
Apart=data[data[column_name[6]]==values_col_6[2]]#Getting the data where the value is Bring people close together
print("The number of people who think A.I will push people further aprt are:")
print(Apart.index.size)
Remaining_people=data.index.size-Apart.index.size#Getting the remaining people
sizes=[Apart.index.size,Remaining_people]#Storing all the responses in an array
colors=['cyan','navy']#Storing all the colors in an array
labesl=[values_col_6[2],'Close together or have no effect']#Storing all the labels in an array
plt.pie(sizes,labels=labesl,colors=colors,autopct='%1.1f%%',startangle=90)
plt.title('A.I will push people further apart  Opinion', fontsize=18, fontweight='bold', color='navy')
plt.legend(loc='best', bbox_to_anchor=(1, 0.8))
plt.axis('equal')
plt.show()
```

Output-


A.I will push people further apart Opinion
Push people further apart
41.3%
58.7%
Close together or have no effect
Push people further apart
Close together or have no effect

l

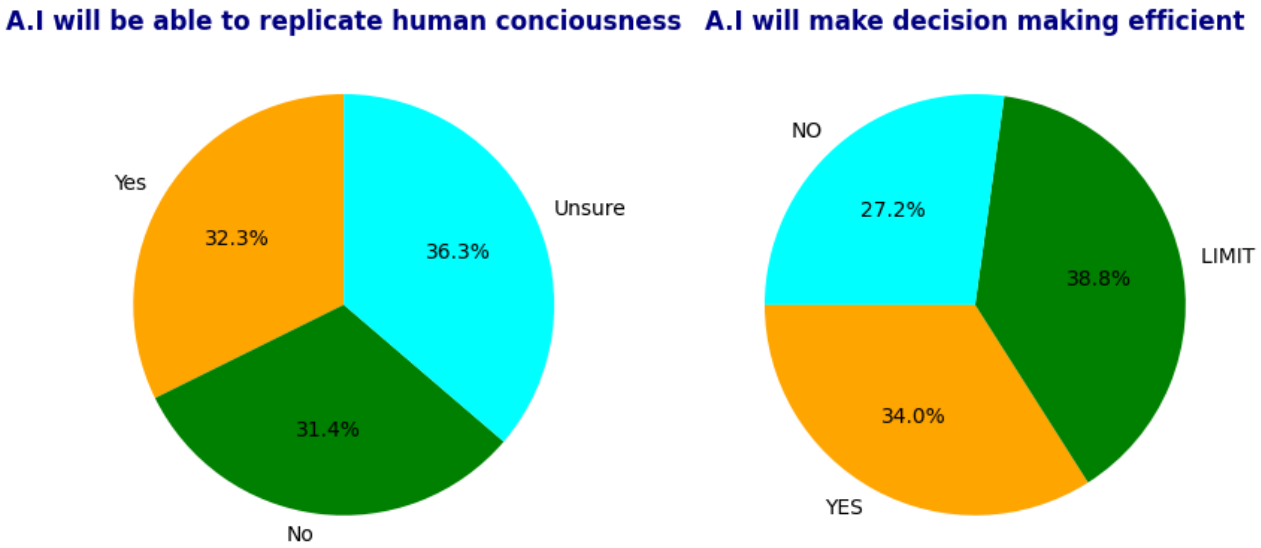
Q6-How many people think that A.I will be able to replicate human emotion and make decision making efficient?

Code-

```
# Q6-How many people think that A.I will be able to replicate human conciousness and make decision making efficient?
values_col_7=data.iloc[:,7].unique()#Getting the unique values of column 7
print(values_col_7)
values_col_8=data.iloc[:,8].unique()#Getting the unique values of column 8
print(values_col_8)
Yes=data[data[column_name[7]]==values_col_7[1]]#Getting the data where the value is Yes
No=data[data[column_name[7]]==values_col_7[2]]#Getting the data where the value is No
Unsure=data[data[column_name[7]]==values_col_7[0]]#Getting the data where the value is Unsure
print(Yes.index.size)
Yes_decision_efficeint=Yes[Yes[column_name[8]]==values_col_8[0]]#Getting the data where the value for efficient decision making with Yes as answer
Yes_Limit_decision_efficeint=Yes[Yes[column_name[8]]==values_col_8[1]]#Getting the data where the value for efficient decision making with No as answer
Yes_No_effective_decision_making=Yes[Yes[column_name[8]]==values_col_8[2]]#Getting the data where the value for efficient decision making with No effect as answer

# Plotting pie chart of People with Yes and No as answer for A.I will be able to replicate human conciousness and make decision making efficient
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
size=[Yes.index.size,No.index.size,Unsure.index.size]#Storing all the responses in an array
```

```
labesl=[values_col_7[1],values_col_7[2],values_col_7[0]]#Storin
g all the labels in an array
colors=['#FFA500','#008000','cyan']#Storing all the colors in an
array
ax[0].pie(size,labels=labesl,colors=colors,autopct='%1.1f%
%',startangle=90)
ax[0].set_title('A.I will be able to replicate human conciousness',
fontsize=12, fontweight='bold', color='navy')

size_new=[Yes_decision_efficeint.index.size,Yes_Limit_decision_
efficeint.index.size,Yes_No_effective_decision_making.index.size
]#Storing all the responses in an array
labesl_new=["YES","LIMIT ","NO"]#Storing all the labels in an
array
colors_new=['#FFA500','#008000','cyan']#Storing all the colors in
an array
ax[1].pie(size_new,labels=labesl_new,colors=colors_new,autopct=
'%1.1f%%',startangle=180)
ax[1].set_title('A.I will make decision making efficient',
fontsize=12, fontweight='bold', color='navy')
```

Output-

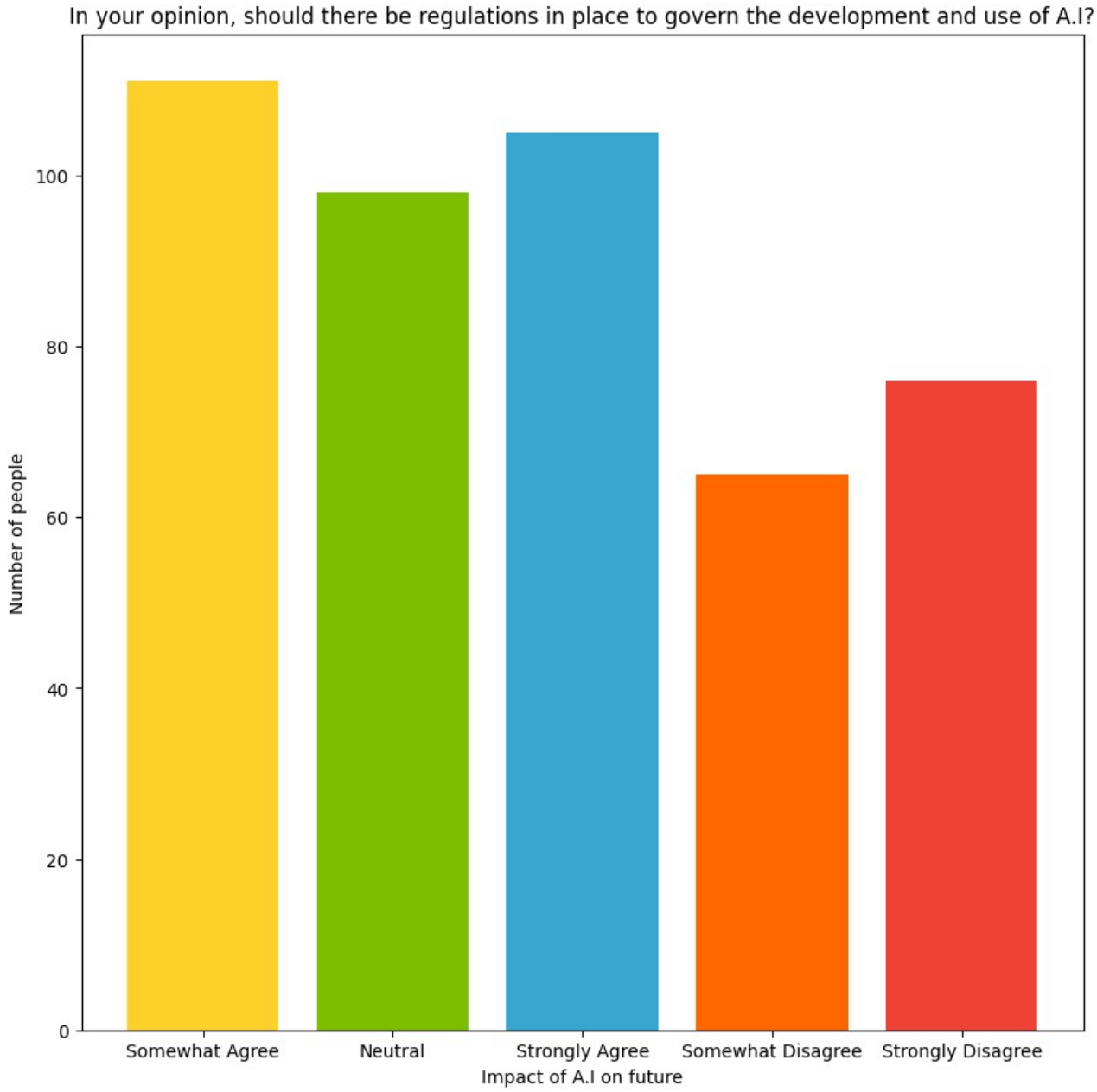


Q7-How many people think that there must be regulations placed on governing and development of A.I

Code-

```
# Q7-How many people think that there must be regulations placed on governing and development of A.I
data.iloc[:,-6]
values=data.iloc[:,-6].unique()#Getting the unique values of the required column
print(type(values))
output=[]
for i in values:
    temp=data.iloc[:,-6].value_counts()[i]
    output.append(temp)

print(output)
output[5]=output[5]+output[3]
output=np.delete(output,3)
values=np.delete(values,3)
#Creating Array of values
print(values)
plt.figure(figsize=(10, 10))
plt.bar(values,output,color = ['#FCD12A', '#7EBE01', '#37A7D1', '#FF6600', '#EF4135'])
plt.ylabel('Number of people')
plt.xlabel('Impact of A.I on future')
plt.title(column_name[-6])
plt.show()
```

Output-


In your opinion, should there be regulations in place to govern the development and use of A.I?
Number of people
0
20
40
60
80
100
Somewhat Agree
Neutral
Strongly Agree
Somewhat Disagree
Strongly Disagree
Impact of A.I on future

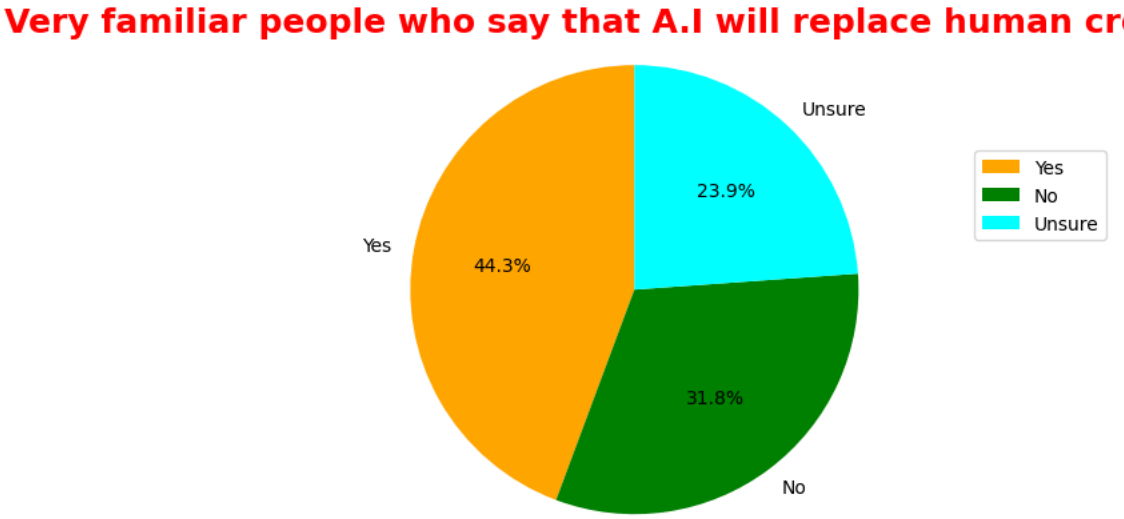
l

Q8-How many people think that A.I will replace human creativity who are very familiar with the concept of A.I

Code-

```
#Q8-How many people think that A.I will replace human creativity who are very familiar with the concept of A.I
value_col_1=data.iloc[:,1].unique()#Getting the unique values of column 1
Very_familiar=data[data[column_name[1]]==value_col_1[1]]#Getting the data where the value is Very familiar
values_bcol_4=data.iloc[:,-4].unique()#Getting the unique values of column 12
print(values_bcol_4)
Createvity_replace=Very_familiar[Very_familiar[column_name[-4]]==values_bcol_4[1]]#Getting the data where the value for efficient decision making with Yes as answer
Not_replace=Very_familiar[Very_familiar[column_name[-4]]==values_bcol_4[0]]#Getting the data where the value for efficient decision making with No as answer
Unsure_replace=Very_familiar[Very_familiar[column_name[-4]]==values_bcol_4[2]]#Getting the data where the value for efficient decision making with Unsure effect as answer
size=[Createvity_replace.index.size,Not_replace.index.size,Unsure_replace.index.size]#Storing all the responses in an array
labels=[values_bcol_4[1],values_bcol_4[0],values_bcol_4[2]]#Storing all the labels in an array
colors=['#FFA500','#008000','cyan']#Storing all the colors in an array
plt.title('Very familiar people who say that A.I will replace human creativity', fontsize=18, fontweight='bold', color='red')
plt.pie(size,labels=labels,colors=colors,autopct='%1.1f%%',startangle=90)
plt.legend(loc='best', bbox_to_anchor=(1, 0.8))
plt.axis('equal')
plt.show()
```

Output-


Very familiar people who say that A.I will replace human creativity
Unsure
23.9%
Yes
44.3%
31.8%
No
Yes
No
Unsure

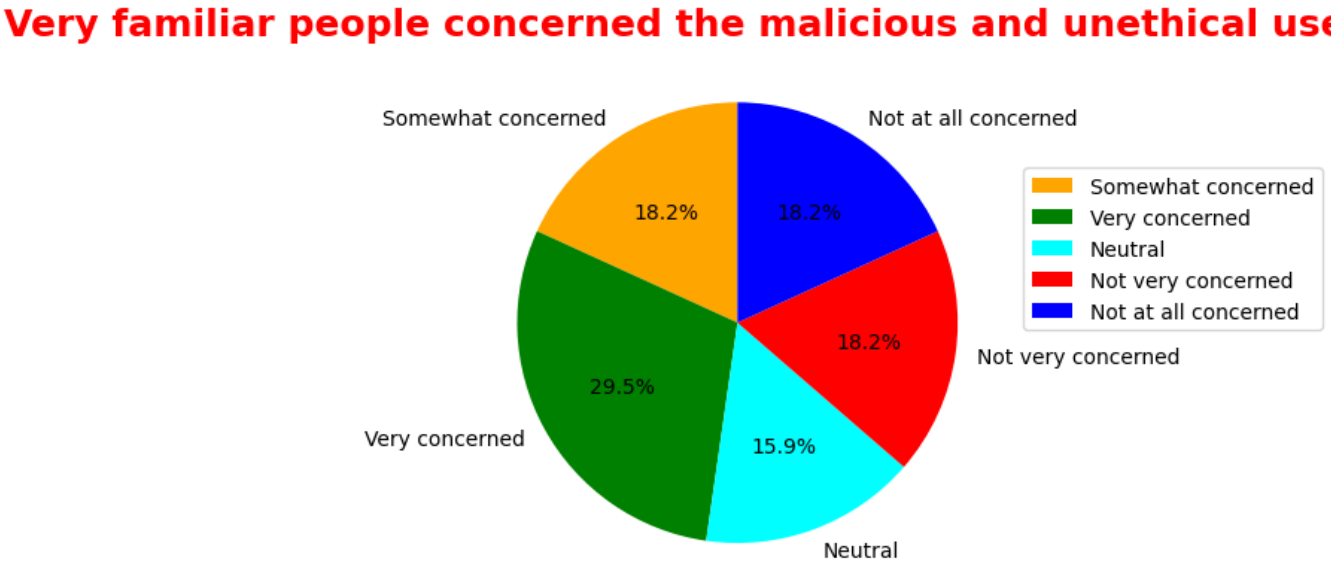
l

Q9-How many people are concerned with the malicious and unethical use of A.I given they are very familiar with the concept of A.I

```
#Q9-How many people are concerned with the malicious and unethical use of A.I given they are very familiar with the concept of A.I
value_col_1=data.iloc[:,1].unique()#Getting the unique values of column 1
Very_familiar=data[data[column_name[1]]==value_col_1[1]]#Getting the data where the value is Very familiar
value_col_13=data.iloc[:,13].unique()#Getting the unique values of column 13
Somewhat_concerned=Very_familiar[Very_familiar[column_name[13]]==value_col_13[0]]#Getting the data where the value for efficient decision making with Somewhat concerned as answer
Very_concerned=Very_familiar[Very_familiar[column_name[13]]==value_col_13[1]]#Getting the data where the value for efficient decision making with Very concerned as answer
Neutral=Very_familiar[Very_familiar[column_name[13]]==value_col_13[2]]#Getting the data where the value for efficient decision making with Neutral as answer
Not_very_concerned=Very_familiar[Very_familiar[column_name[13]]==value_col_13[3]]#Getting the data where the value for efficient decision making with Not very concerned as answer
Not_at_all_concerned=Very_familiar[Very_familiar[column_name[13]]==value_col_13[4]]#Getting the data where the value for efficient decision making with Not at all concerned as answer
size=[Somewhat_concerned.index.size,Very_concerned.index.size,Neutral.index.size,Not_very_concerned.index.size,Not_at_all_concerned.index.size]#Storing all the responses in an array
labesl=[value_col_13[0],value_col_13[1],value_col_13[2],value_col_13[3],value_col_13[4]]#Storing all the labels in an array
colors=['#FFA500','#008000','cyan','red','blue']#Storing all the colors in an array
```

```
plt.title('Very familiar people concerned the malicious and
unethical use of A.I', fontsize=18, fontweight='bold', color='red')
plt.pie(size,labels=labesl,colors=colors,autopct='%1.1f%
%',startangle=90)
plt.legend(loc='best', bbox_to_anchor=(1, 0.8))
plt.show()
```

Output-



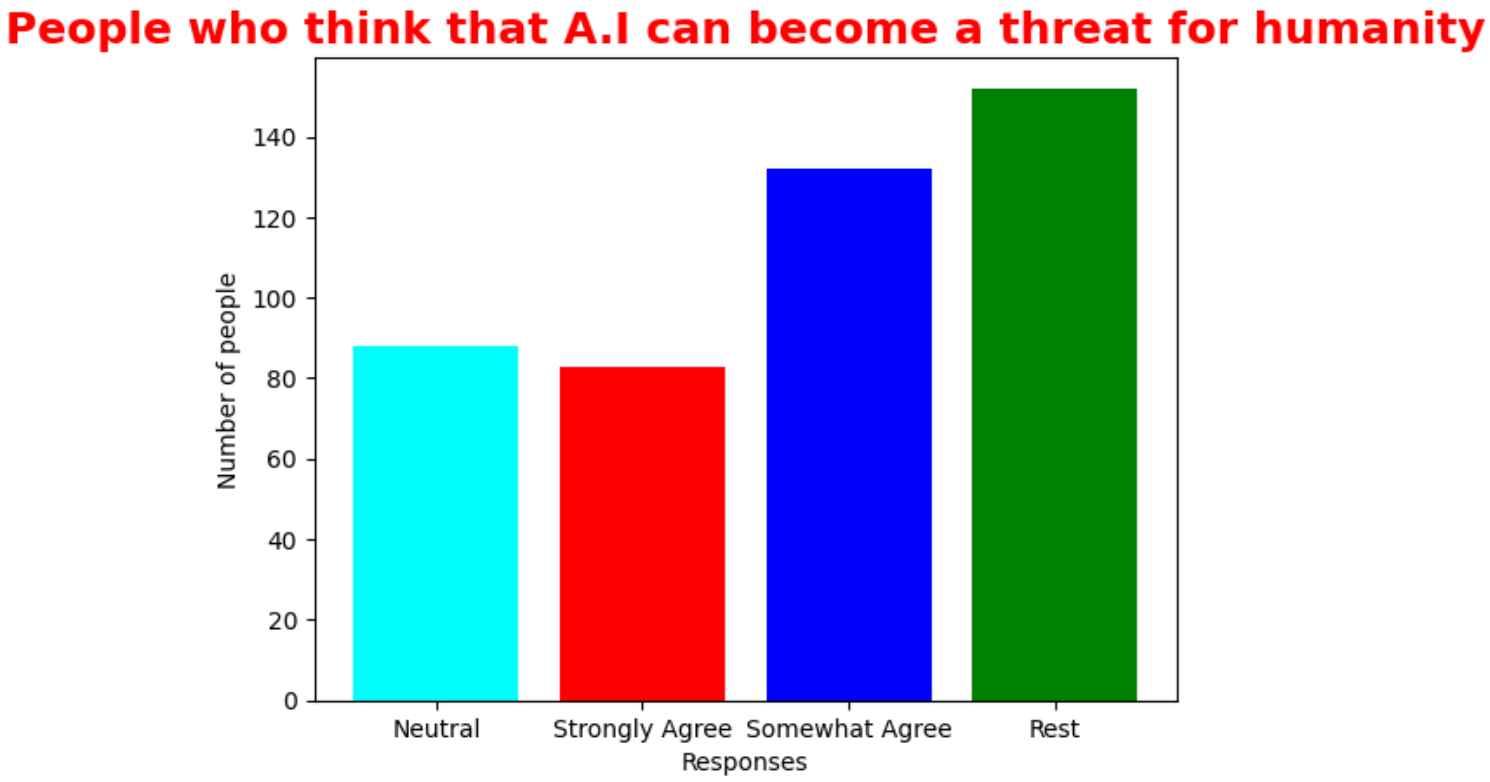
l

Q10-How many people think that A.I can become a threat for humanity.

Code-

```
#Q10-How many people think that A.I can become a threat for humanity.
value_col_14=data.iloc[:,14].unique()#Getting the unique values of column 13
print(value_col_14)
print(column_name[14])
Neutral=data[data[column_name[14]]==value_col_14[0]]#Getting the data where the value for efficient decision making with Neutral as answer
Strongly_agree=data[data[column_name[14]]==value_col_14[1]]#Getting the data where the value for efficient decision making with Strongly agree as answer
Somewhat_agree=data[data[column_name[14]]==value_col_14[2]]#Getting the data where the value for efficient decision making with Somewhat agree as answer
Rest=455-(Neutral.index.size+Strongly_agree.index.size+Somewhat_agree.index.size)#Getting the Rest answer value
size=[Neutral.index.size,Strongly_agree.index.size,Somewhat_agree.index.size,Rest]#Storing all the responses in an array
labesl=[value_col_14[0],value_col_14[1],value_col_14[2],'Rest']#Storing all the labels in an array
colors=['cyan','red','blue','green']#Storing all the colors in an array
plt.bar(labesl,size,color=colors)
plt.title('People who think that A.I can become a threat for humanity', fontsize=18, fontweight='bold', color='red')
plt.ylabel('Number of people')
plt.xlabel('Responses')
plt.show()
```

Output-


People who think that A.I can become a threat for humanity
Number of people
0
20
40
60
80
100
120
140
Neutral
Strongly Agree
Somewhat Agree
Rest
Responses

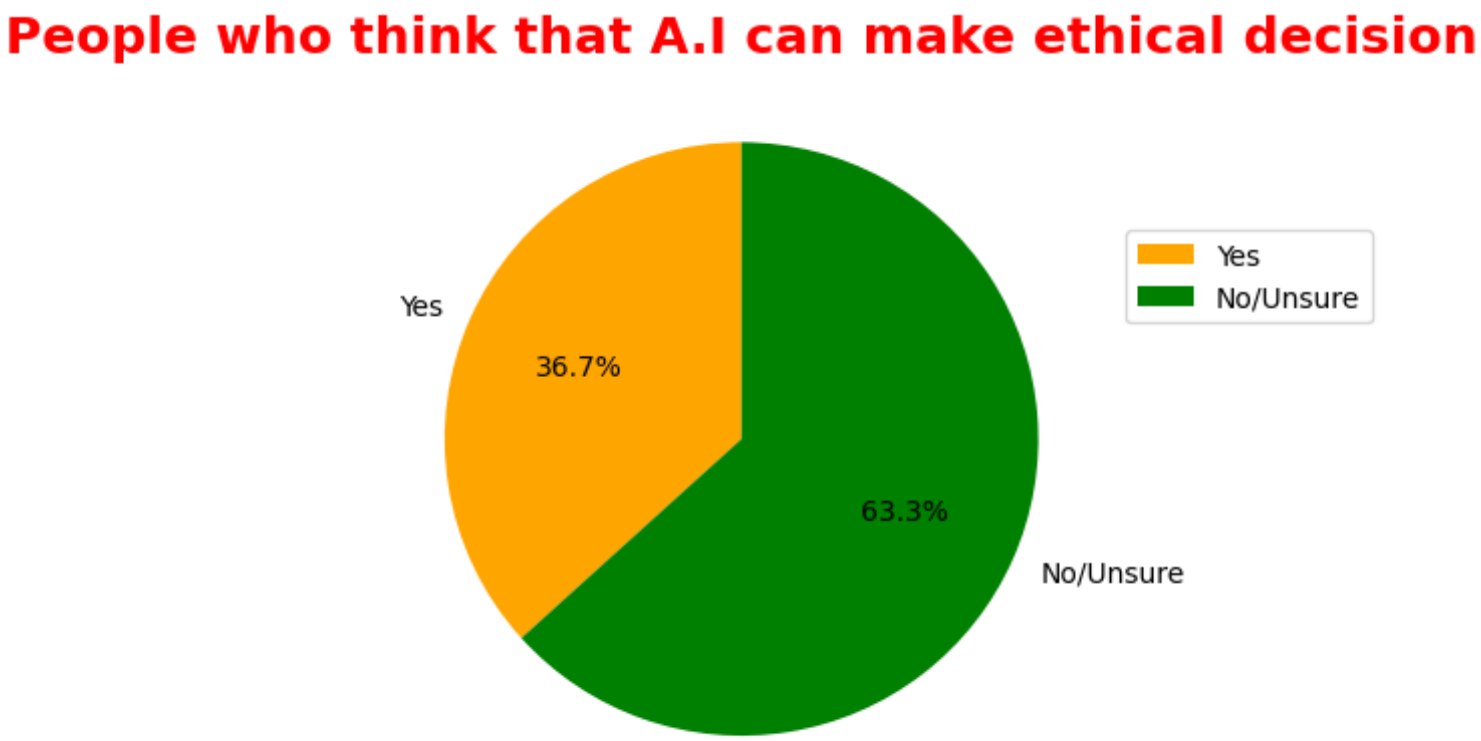
l

Q11-How many people think that A.I can make ethical decision?

Code-

```
# Q11-How many people think that A.I can make ethical decision?
values_col_7=data.iloc[:,-7].unique()#Getting the unique values of column
print(values_col_7)
Yes=data[data[column_name[-7]]==values_col_7[2]]#Getting the data where the value for efficient decision making with Yes as answer
No_Unsure=data[data[column_name[-7]]!=values_col_7[2]]#Getting the data where the value for efficient decision making with No/Unsure as answer
size=[Yes.index.size,No_Unsure.index.size]#Storing all the responses in an array
labels=[values_col_7[2],'No/Unsure']#Storing all the labels in an array
colors=['#FFA500','#008000']#Storing all the colors in an array
plt.title('People who think that A.I can make ethical decision', fontsize=18, fontweight='bold', color='red')

plt.pie(size,labels=labels,colors=colors,autopct='%1.1f%%',startangle=90)
plt.legend(loc='best', bbox_to_anchor=(1, 0.8))
plt.show()
```

Output-


People who think that A.I can make ethical decision
Yes
36.7%
63.3%
No/Unsure
Yes
No/Unsure

l

Q12-What is the most impacted sector by A.I according to people who have different familarity with A.I (i.e Strong familarity,Somewhat familarity,No familarity.)

Code-

```
#Q12-What is the most impacted sector by A.I according to people who have different familarity with A.I (i.e Strong familarity,Somewhat familarity,No familarity.)
value_col_1
print(value_col_1)
Somewhat_familiar=data[data[column_name[1]]==value_col_1[0]]#Getting the data where the value is Somewhat familiar
Very_familiar=data[data[column_name[1]]==value_col_1[1]]#Getting the data where the value is Very familiar
Not_familiar=data[data[column_name[1]]==value_col_1[2]]#Getting the data where the value is Not familiar
value_col_5=data.iloc[:,5].unique()#Getting the unique values of column 5
Count_Somewhat_familiar
=Somewhat_familiar[column_name[5]].value_counts()
print("People who are Somewhat familiar with A.I think these will the most impacted sector by A.I")
print(Count_Somewhat_familiar)
Count_very_familiar=Very_familiar[column_name[5]].value_counts()
print("People who are Very familiar with A.I think these will the most impacted sector by A.I")
print(Count_very_familiar)
Count_Not_familiar=Not_familiar[column_name[5]].value_counts()
print("People who are Not familiar with A.I think these will the most impacted sector by A.I")
print(Count_Not_familiar)
```

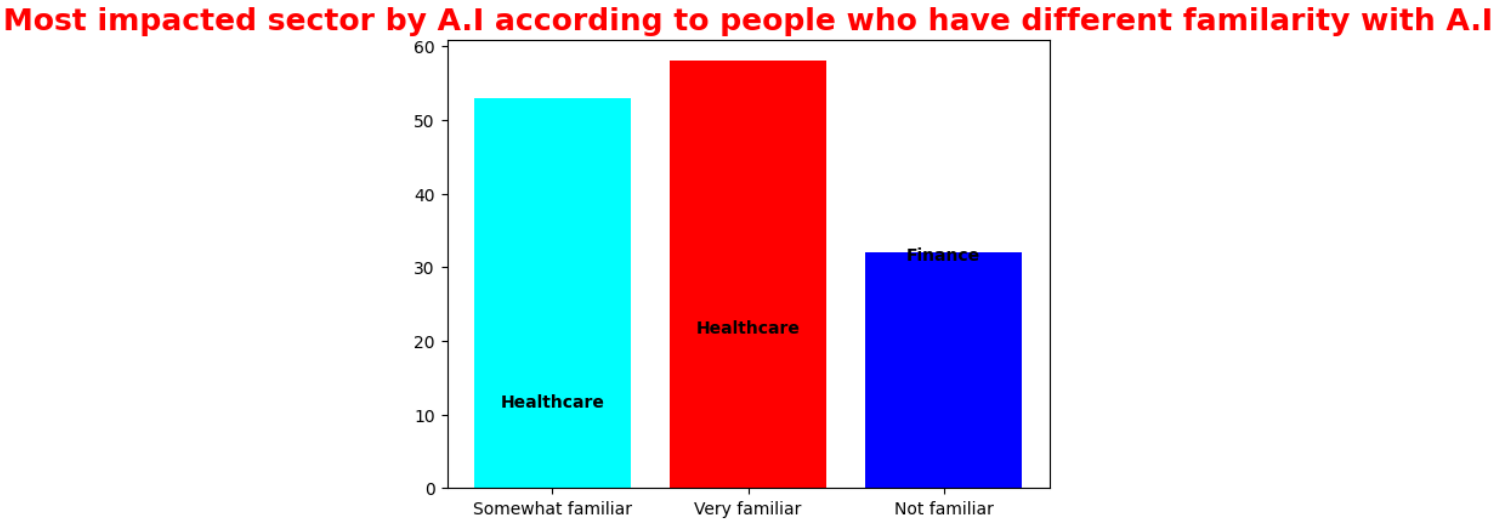
```
most_frequent_value=[Count_Somewhat_familiar.idxmax(),Count
_very_familiar.idxmax(),Count_Not_familiar.idxmax()]
print(most_frequent_value)
fig , ax=plt.subplots()

size=[Count_Somewhat_familiar.max(),Count_very_familiar.max(
),Count_Not_familiar.max()]
labels=['Somewhat familiar','Very familiar','Not familiar']
ax.bar(labels,size,color=['cyan','red','blue'])
plt.title('Most impacted sector by A.I according to people who
have different familarity with A.I', fontsize=18, fontweight='bold',
color='red')
k=1
for i, v in enumerate(y):
    ax.text(i, v+1, most_frequent_value[k-1], ha='center',
fontweight='bold',color='black')
    k=k+1
    if(k>len(most_frequent_value)):
        break

plt.show()
```

Output-

```
['Somewhat famililar' 'Very familiar' 'Not familiar at all']
People who are Somewhat familiar with A.I think these will the most impacted sector by A.I
Healthcare                          53
Manufacturing                       50
Finance                             35
Retail                              28
In allmost areas.                    1
Software                             1
Software as a Service Companies      1
Marketing                            1
Everything                           1
Name: In what industries do you think A.I will have the most impact?, dtype: int64
People who are Very familiar with A.I think these will the most impacted sector by A.I
Healthcare                            58
Manufacturing                         49
Retail                                32
Finance                               32
IT sector                              2
Tech                                   1
Anupama Kumari                         1
All these option will have impact      1
Name: In what industries do you think A.I will have the most impact?, dtype: int64
People who are Not familiar with A.I think these will the most impacted sector by A.I
Finance          32
Retail           28
Manufacturing    24
Healthcare       24
Name: In what industries do you think A.I will have the most impact?, dtype: int64
['Healthcare', 'Healthcare', 'Finance']
```

l

Graph-


Most impacted sector by A.I according to people who have different familarity with A.I
60
50
40
30
20
10
0
Healthcare
Healthcare
Finance
Somewhat familiar
Very familiar
Not familiar

l

Q13-Are people with Strong familarity with A.I giving similar answers?

Answer-
#Are people with Strong familarity with A.I giving similar answers?
#Yes we can say that people with Strong familarity with A.I giving similar answers based on all the questions that we have the answer for till now.

Q14-Can we classify all the responses of people based on there familarity with A.I and come up with a conclusion for this report?

Code-

```
#Can we classify all the responses of people based on there familarity with A.I and come up with a conclusion for this report

#We can do this by checking the last answer of the survey and classify it based on familarity
print(column_name[15])
value_col_15=data.iloc[:,15].unique()#Getting the unique values of column 15
print(value_col_15)
Very_familiar=data[data[column_name[1]]==value_col_1[0]]#Getting the data where the value is Very familiar
Somewhat_familiar=data[data[column_name[1]]==value_col_1[1]]#Getting the data where the value is Somewhat familiar
Not_familiar=data[data[column_name[1]]==value_col_1[2]]#Getting the data where the value is Not familiar
Very_familiar_Weak_economy=Very_familiar[Very_familiar[column_name[15]]==value_col_15[1]]#Getting the data where the value is Weak economy
Very_familiar_Improve_economy=Very_familiar[Very_familiar[column_name[15]]==value_col_15[0]]#Getting the data where the value is Improve economy
Very_familiar_No_effect_economy=Very_familiar[Very_familiar[column_name[15]]==value_col_15[2]]#Getting the data where the value is No effect economy
size1=[Very_familiar_Weak_economy.index.size,Very_familiar_Improve_economy.index.size,Very_familiar_No_effect_economy.index.size]
Somewhat_familiar_weak_economy=Somewhat_familiar[Somewhat_familiar[column_name[15]]==value_col_15[1]]#Getting the data where the value is Weak economy
```

```
Somewhat_familiar_Improve_economy=Somewhat_familiar[Somewhat_familiar[column_name[15]]==value_col_15[0]]#Getting the data where the value is Improve economy
Somewhat_familiar_No_effect_economy=Somewhat_familiar[Somewhat_familiar[column_name[15]]==value_col_15[2]]#Getting the data where the value is No effect economy
size2=[Somewhat_familiar_weak_economy.index.size,Somewhat_familiar_Improve_economy.index.size,Somewhat_familiar_No_effect_economy.index.size]
Not_familiar_weak_economy=Not_familiar[Not_familiar[column_name[15]]==value_col_15[1]]#Getting the data where the value is Weak economy
Not_familiar_Improve_economy=Not_familiar[Not_familiar[column_name[15]]==value_col_15[0]]#Getting the data where the value is Improve economy
Not_familiar_No_effect_economy=Not_familiar[Not_familiar[column_name[15]]==value_col_15[2]]#Getting the data where the value is No effect economy
size3=[Not_familiar_weak_economy.index.size,Not_familiar_Improve_economy.index.size,Not_familiar_No_effect_economy.index.size]
bar_width = 0.25
x_labels = ['Weaken Economy', 'Improve Economy', 'No Effect']
fig, ax = plt.subplots()
ax.bar(np.arange(len(size1)), size1, width=bar_width, color='b', align='center', label='Strong Familiarity')
ax.bar(np.arange(len(size2))+bar_width, size2, width=bar_width, color='g', align='center', label='Somewhat Familiarity')
ax.bar(np.arange(len(size3))+bar_width*2, size3, width=bar_width, color='r', align='center', label='No Familiarity')
ax.set_title('Impact of A.I on Economy && Classification', fontsize=14)
ax.set_xlabel('Sectors', fontsize=12)
ax.set_ylabel('Number of people', fontsize=12)
ax.set_xticks(np.arange(len(x_labels))+bar_width)
```

```
ax.set_xticklabels(x_labels)
ax.legend()
plt.show()
```

Output-