In [81]:
```
#Imported the necessary packages for analysis
import pandas as pd
import numpy as np
import math as mat
import matplotlib.pyplot as plt
import seaborn as sns

#Loaded the dataset
jobdf= pd.read_csv("C:/Users/ADMIN/Downloads/ai_job_dataset.csv")
jobdf.head(20)
```

Out[81]:

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_typ
0	AI00001	AI Research Scientist	90376	USD	SE	C
1	AI00002	AI Software Engineer	61895	USD	EN	C
2	AI00003	AI Specialist	152626	USD	MI	
3	AI00004	NLP Engineer	80215	USD	SE	
4	AI00005	AI Consultant	54624	EUR	EN	I
5	AI00006	AI Architect	123574	EUR	SE	C
6	AI00007	Principal Data Scientist	79670	GBP	MI	
7	AI00008	NLP Engineer	70640	EUR	EN	
8	AI00009	Data Analyst	160710	USD	SE	C
9	AI00010	AI Software Engineer	102557	USD	SE	I
10	AI00011	Autonomous Systems Engineer	102322	USD	SE	I
11	AI00012	AI Architect	115047	USD	EX	C
12	AI00013	AI Consultant	124355	EUR	SE	C
13	AI00014	Autonomous Systems Engineer	68760	USD	EN	C

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_typ
14	AI00015	AI Research Scientist	150122	USD	SE	
15	AI00016	AI Product Manager	78846	GBP	EN	I
16	AI00017	Principal Data Scientist	59823	USD	EN	
17	AI00018	Machine Learning Engineer	181139	EUR	EX	C
18	AI00019	Data Engineer	155300	USD	SE	C
19	AI00020	Research Scientist	93851	EUR	MI	I

In [82]:
```
#Filtered the dataset to get remote workers
remote_jobbers= jobdf.loc[(jobdf['company_location'] != jobdf['employee_residenc
remote_jobbers.head(10)
```

Out[82]:

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_typ
1	AI00002	AI Software Engineer	61895	USD	EN	C
4	AI00005	AI Consultant	54624	EUR	EN	P
19	AI00020	Research Scientist	93851	EUR	MI	P
23	AI00024	AI Product Manager	52167	USD	MI	F
70	AI00071	NLP Engineer	51907	USD	MI	F
74	AI00075	Data Engineer	51920	USD	EN	P
115	AI00116	NLP Engineer	61055	USD	EN	F
119	AI00120	AI Product Manager	106633	USD	SE	F
136	AI00137	Robotics Engineer	172916	USD	SE	P
146	AI00147	AI Research Scientist	224919	USD	EX	F

In [83]:
```
#Filtered the dataset to get hybrid workers
hybrid_jobbers= jobdf.loc[(jobdf['company_location'] != jobdf['employee_residenc
hybrid_jobbers.head(10)
```

Out[83]:

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_ty
26	AI00027	ML Ops Engineer	80979	USD	MI	
45	AI00046	AI Research Scientist	174663	USD	SE	
49	AI00050	AI Consultant	124871	USD	EX	
63	AI00064	AI Software Engineer	99972	USD	MI	
86	AI00087	Autonomous Systems Engineer	33314	USD	EN	
98	AI00099	Machine Learning Researcher	290199	USD	EX	
106	AI00107	AI Research Scientist	152658	USD	EX	
121	AI00122	Principal Data Scientist	64531	USD	EN	
139	AI00140	Principal Data Scientist	122598	USD	SE	
153	AI00154	AI Research Scientist	92445	EUR	MI	

In [84]:
```
#Filtered the dataset to get on-site workers
noremote_jobbers= jobdf.loc[(jobdf['company_location'] != jobdf['employee_reside
noremote_jobbers.head(10)
```

Out[84]:

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_type
2	AI00003	AI Specialist	152626	USD	MI	FL
15	AI00016	AI Product Manager	78846	GBP	EN	PT
20	AI00021	Data Engineer	134197	USD	MI	FT
27	AI00028	Data Analyst	52997	USD	MI	PT
34	AI00035	Deep Learning Engineer	150864	USD	SE	FL
35	AI00036	Head of AI	126942	USD	EX	CT
40	AI00041	Data Scientist	96956	USD	MI	FT
59	AI00060	AI Software Engineer	110222	GBP	SE	FL
67	AI00068	Machine Learning Engineer	316182	USD	EX	PT
130	AI00131	AI Specialist	131756	EUR	SE	PT

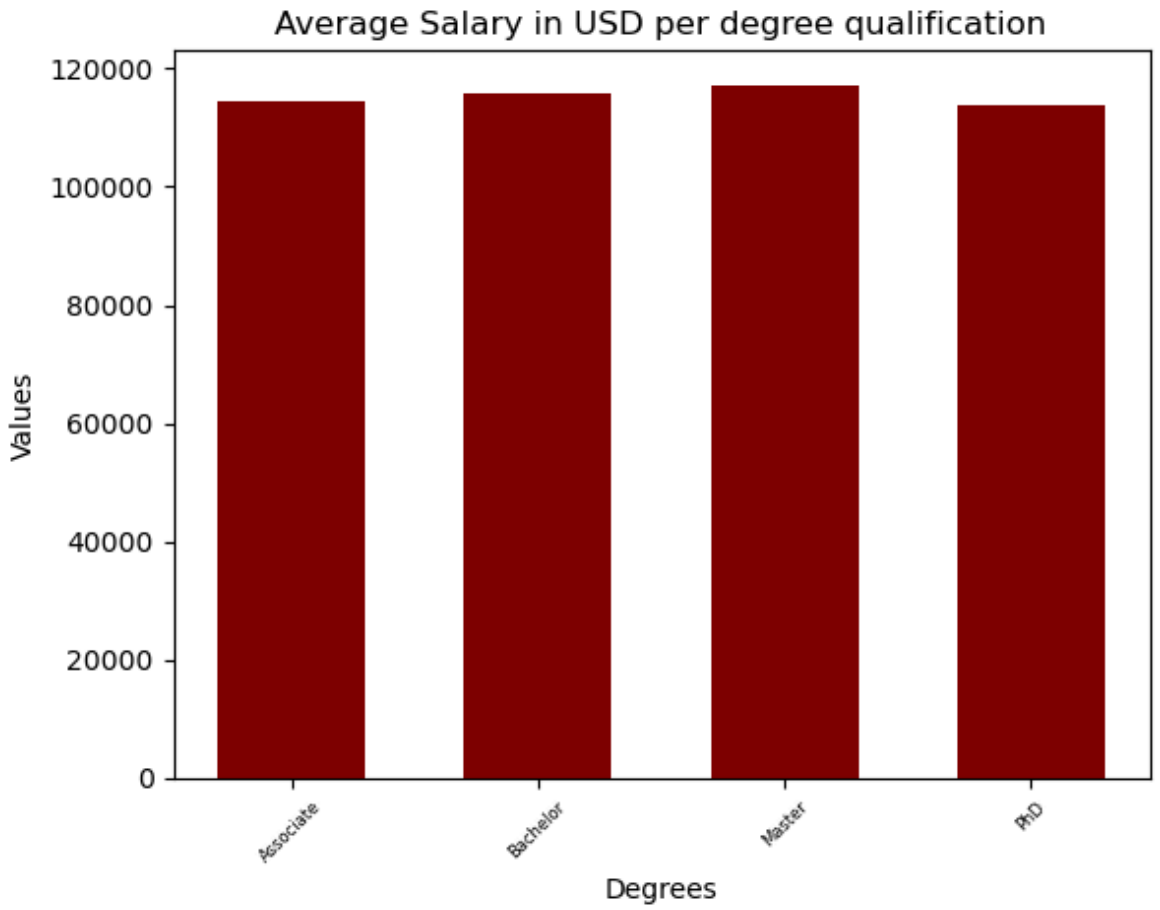
In [85]:
```
#Created a dataframe to contain evrage salary per degree
average_salary_per_educationreq= jobdf.groupby('education_required')['salary_usd

average_salary_per_educationreq.rename(columns={'education_required': 'Degree'},
average_salary_per_educationreq.rename(columns={'salary_usd': 'Average Salary in
average_salary_per_educationreq.head(4)
```

Out[85]:

	Degree	Average Salary in USD
0	Associate	114605.708058
1	Bachelor	115861.629190
2	Master	117171.815902
3	PhD	113728.165579

In [86]:
```
#Created a bar chart to visualize the data
plt.bar(average_salary_per_educationreq['Degree'], average_salary_per_educationr
plt.xlabel('Degrees')
plt.ylabel('Values')
plt.xticks(fontsize=6, rotation=45)
plt.title('Average Salary in USD per degree qualification')
```

Out[86]:
```
Text(0.5, 1.0, 'Average Salary in USD per degree qualification')
```



In [87]:
```
#Created a dataframe to contain evrage salary per job type
average_salary_per_jobtype= jobdf.groupby('employment_type')['salary_usd'].mean(

average_salary_per_jobtype.rename(columns={'employment_type': 'Job Type'}, inpla
average_salary_per_jobtype.rename(columns={'salary_usd': 'Average Salary in USD'
average_salary_per_jobtype.head(100)
```

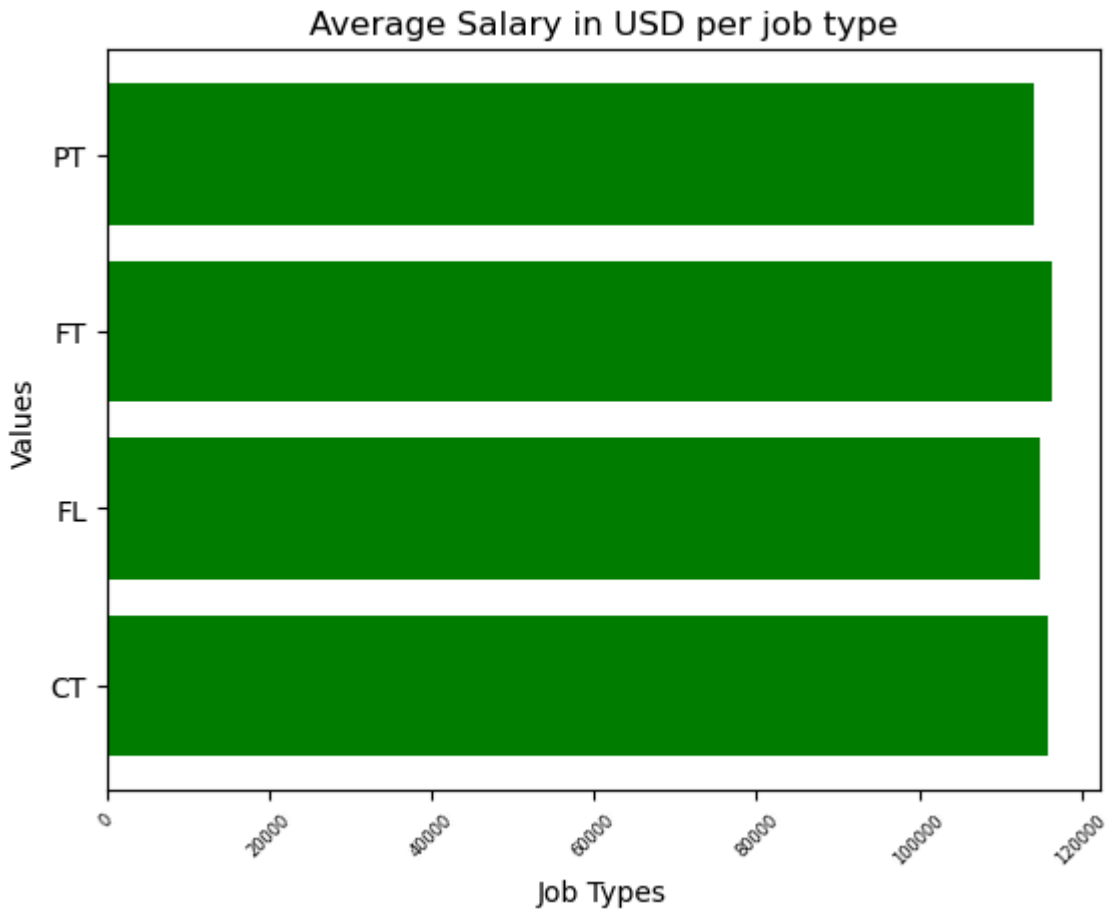
Out[87]:

	Job Type	Average Salary in USD
0	CT	115918.919645
1	FL	114967.645290
2	FT	116338.137723
3	PT	114146.881909

In [88]:
```
#Created a horizontal bar chart to visualize the data
plt.barh(average_salary_per_jobtype['Job Type'], average_salary_per_jobtype['Ave
plt.xlabel('Job Types')
plt.ylabel('Values')
```

```
plt.xticks(fontsize=6, rotation=45)
plt.title('Average Salary in USD per job type')
```

Out[88]: Text(0.5, 1.0, 'Average Salary in USD per job type')

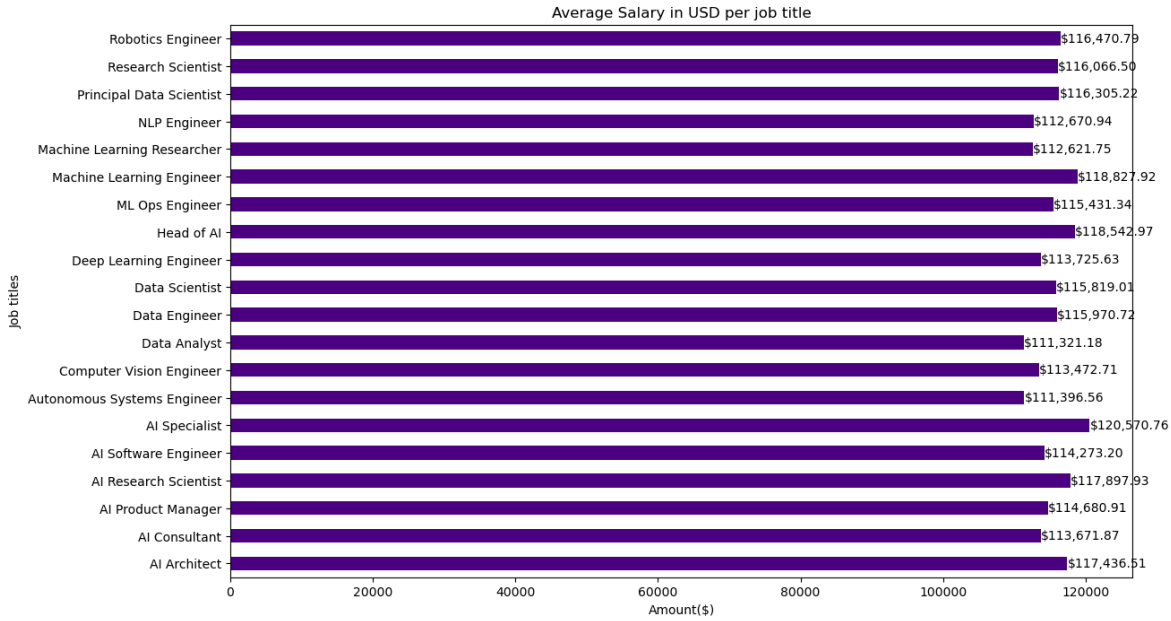


In [89]:
```
#Created a dataframe to contain evrage salary per job title
average_salary_per_jobtitle= jobdf.groupby('job_title')['salary_usd'].mean().res

average_salary_per_jobtitle.rename(columns={'job_title': 'Job Title'}, inplace=
average_salary_per_jobtitle.rename(columns={'salary_usd': 'Average Salary in USD
average_salary_per_jobtitle.head(100)
```

Out[89]:

	Job Title	Average Salary in USD
0	AI Architect	117436.513619
1	AI Consultant	113671.870739
2	AI Product Manager	114680.909825
3	AI Research Scientist	117897.925926
4	AI Software Engineer	114273.201531
5	AI Specialist	120570.758242
6	Autonomous Systems Engineer	111396.557272
7	Computer Vision Engineer	113472.707182
8	Data Analyst	111321.180501
9	Data Engineer	115970.720961
10	Data Scientist	115819.008333
11	Deep Learning Engineer	113725.632312
12	Head of AI	118542.968627
13	ML Ops Engineer	115431.335172
14	Machine Learning Engineer	118827.919689
15	Machine Learning Researcher	112621.747525
16	NLP Engineer	112670.937008
17	Principal Data Scientist	116305.219346
18	Research Scientist	116066.502695
19	Robotics Engineer	116470.793149

In [90]:
```
#Created a horizontal bar chart to visualize the data
ax = average_salary_per_jobtitle.plot(kind='barh',x='Job Title', y='Average Sala
ax.bar_label(ax.containers[0], label_type='edge', fmt='${:,.2f}')
ax.margins(y=0.1)
```

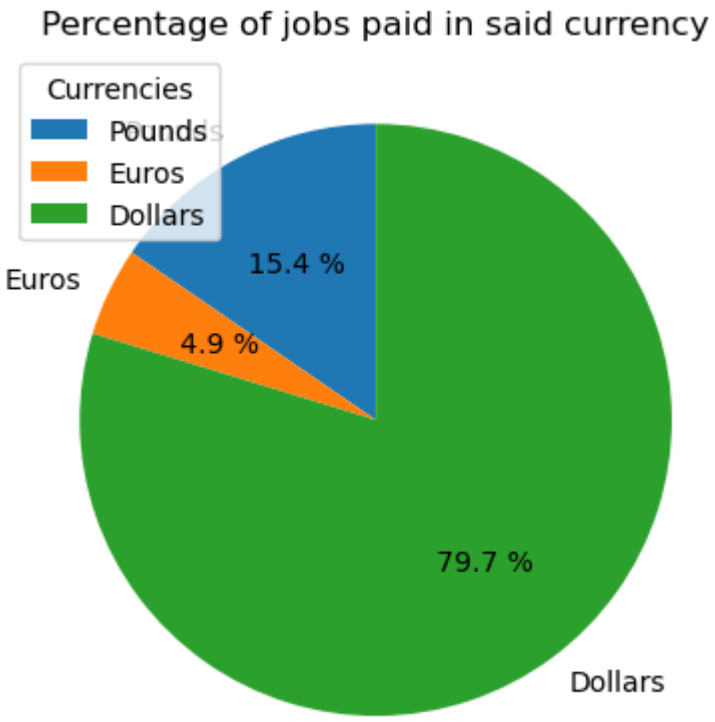


In [91]:
```
#Created a dataframe to contain total number of jobs paid in different currencie
total_salary_per_currency= jobdf.groupby('salary_currency')['job_id'].count().re

total_salary_per_currency.rename(columns={'salary_currency': 'Salary Currency'},
total_salary_per_currency.rename(columns={'job_id': 'Total jobs paid in said cur
total_salary_per_currency.head(100)
```

Out[91]:

	Salary Currency	Total jobs paid in said currency
0	EUR	2314
1	GBP	729
2	USD	11957

In [92]:
```
#Created a pie chart to visualize the data
plt.pie(total_salary_per_currency['Total jobs paid in said currency'], labels =

autopct ='% 1.1f %%', startangle=90)
plt.legend(title= "Currencies")
plt.title("Percentage of jobs paid in said currency")
plt.show()
```

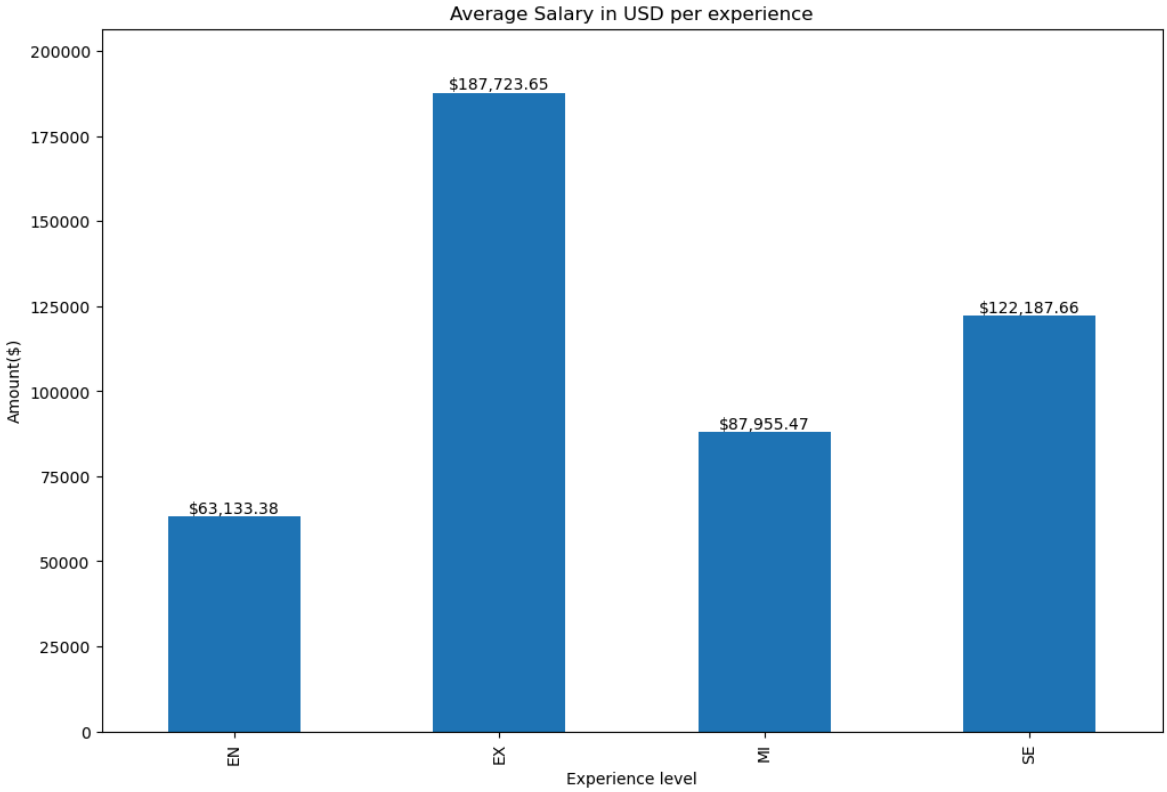


In [93]:
```
#Created a dataframe to contain average salary per experience level
total_experiencel_per_remoter= jobdf.groupby('experience_level')['salary_usd'].m

total_experiencel_per_remoter.rename(columns={'experience_level': 'Experience'},
total_experiencel_per_remoter.rename(columns={'salary_usd': 'Average Salary in U
total_experiencel_per_remoter.head(100)
```

Out[93]:

	Experience	Average Salary in USD
0	EN	63133.377084
1	EX	187723.647340
2	MI	87955.471833
3	SE	122187.657845

In [94]:
```
#Created a bar chart to visualize the data
ax = total_experiencel_per_remoter.plot(kind='bar',x='Experience', y='Average Sa
ax.bar_label(ax.containers[0], label_type='edge', fmt='${:,.2f}')
ax.margins(y=0.1)
```

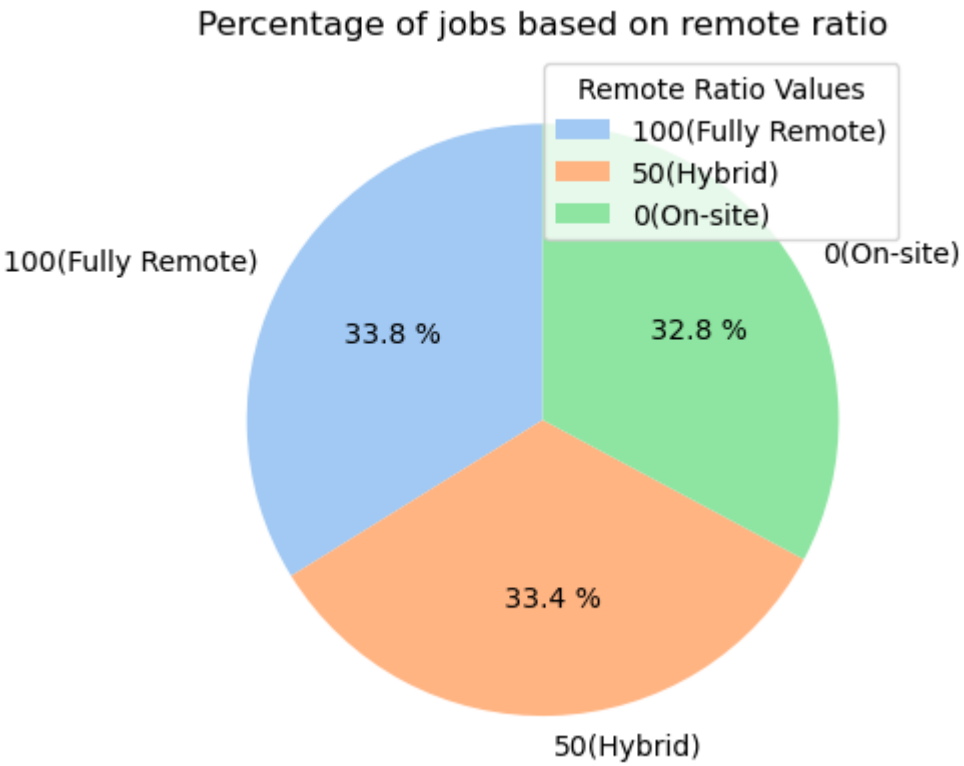


In [95]:
```
#Created a dataframe to contain total number of jobs per remote ratio
total_jobs_per_remoter1= jobdf.groupby('remote_ratio')['experience_level'].count

total_jobs_per_remoter1.rename(columns={'experience_level': 'Total jobs'}, inpla
total_jobs_per_remoter1.rename(columns={'remote_ratio': 'Remote ratio'}, inplace
total_jobs_per_remoter1.head(100)
```

Out[95]:

	Remote ratio	Total jobs
0	0	5075
1	50	5005
2	100	4920

In [96]:
```
#Created a pie chart to visualize the data
plt.pie(total_jobs_per_remoter1['Total jobs'], labels = {"0(On-site)", "50(Hybri
plt.legend(title= "Remote Ratio Values", loc="upper right")
plt.title("Percentage of jobs based on remote ratio")
plt.show()
```



In [97]:
```
#Created a dataframe to contain total number of on-site jobs per industry
noremote_perindust= noremote_jobbers.groupby('industry')['job_id'].count().reset

noremote_perindust.rename(columns={'industry': 'Industry'}, inplace= True)
noremote_perindust.rename(columns={'job_id': 'Total jobs'}, inplace= True)
noremote_perindust.head(100)
```

Out[97]:

	Industry	Total jobs
0	Automotive	75
1	Consulting	102
2	Education	91
3	Energy	96
4	Finance	94
5	Gaming	99
6	Government	95
7	Healthcare	95
8	Manufacturing	89
9	Media	108
10	Real Estate	88
11	Retail	101
12	Technology	98
13	Telecommunications	106
14	Transportation	98

In [98]:

```
#Created a dataframe to contain total number of remote jobs per industry
remote_perindust= remote_jobbers.groupby('industry')['job_id'].count().reset_ind

remote_perindust.rename(columns={'industry': 'Industry'}, inplace= True)
remote_perindust.rename(columns={'job_id': 'Total jobs'}, inplace= True)
remote_perindust.head(100)
```

Out[98]:

	Industry	Total jobs
0	Automotive	95
1	Consulting	95
2	Education	92
3	Energy	91
4	Finance	95
5	Gaming	82
6	Government	71
7	Healthcare	96
8	Manufacturing	88
9	Media	102
10	Real Estate	90
11	Retail	102
12	Technology	87
13	Telecommunications	88
14	Transportation	87

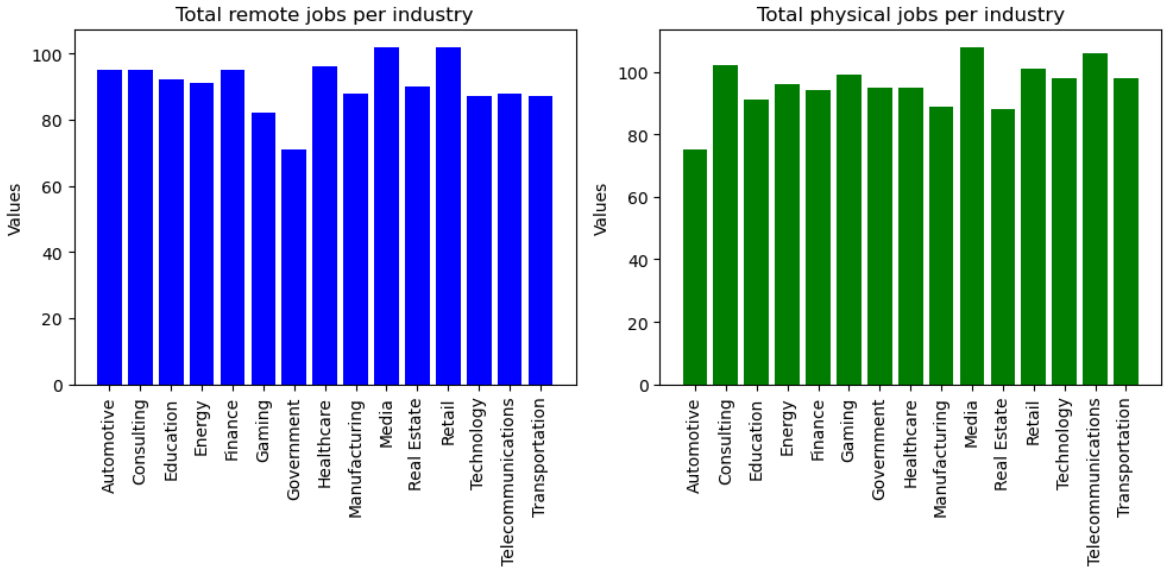
In [99]:
```
#Merged the previous dataframes together
remotevsphys_indust = pd.merge(remote_perindust, noremote_perindust, on= "Indust
remotevsphys_indust.rename(columns={'Total jobs_x': 'Total Remote Jobs'}, inplac
remotevsphys_indust.rename(columns={'Total jobs_y': 'Total Physical jobs'}, inpl
remotevsphys_indust.head(15)
```

Out[99]:

	Industry	Total Remote Jobs	Total Physical jobs
0	Automotive	95	75
1	Consulting	95	102
2	Education	92	91
3	Energy	91	96
4	Finance	95	94
5	Gaming	82	99
6	Government	71	95
7	Healthcare	96	95
8	Manufacturing	88	89
9	Media	102	108
10	Real Estate	90	88
11	Retail	102	101
12	Technology	87	98
13	Telecommunications	88	106
14	Transportation	87	98

In [100…

```
#Created a bar chart to visualize the data
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# First dataset
axes[0].bar(remotevsphys_indust["Industry"],remotevsphys_indust["Total Remote Jo
axes[0].tick_params(axis='x', labelrotation=90)
axes[0].set_title("Total remote jobs per industry")
axes[0].set_ylabel("Values")

# Second dataset
axes[1].bar(remotevsphys_indust["Industry"], remotevsphys_indust["Total Physical
axes[1].tick_params(axis='x', labelrotation=90)
axes[1].set_title("Total physical jobs per industry")
axes[1].set_ylabel("Values")

plt.tight_layout()
plt.show()
```



In [101…

```
#Created a dataframe to contain total number of hybrid jobs per industry
hybrid_perindust= hybrid_jobbers.groupby('industry')['job_id'].count().reset_ind

hybrid_perindust.rename(columns={'industry': 'Industry'}, inplace= True)
hybrid_perindust.rename(columns={'job_id': 'Total Hybrid jobs'}, inplace= True)
hybrid_perindust.head(100)
```

Out[101…

	Industry	Total Hybrid jobs
0	Automotive	96
1	Consulting	115
2	Education	78
3	Energy	107
4	Finance	106
5	Gaming	90
6	Government	85
7	Healthcare	99
8	Manufacturing	96
9	Media	86
10	Real Estate	104
11	Retail	97
12	Technology	97
13	Telecommunications	100
14	Transportation	104

In [102…

```
#Created a bar chart to visualize the data
plt.bar(hybrid_perindust['Industry'], hybrid_perindust['Total Hybrid jobs'], col
plt.xlabel('Industry')
plt.ylabel('Values')
```

```
plt.xticks(fontsize=10, rotation=90)
plt.title('Total Hybrid jobs per industry')
```

Out[102… Text(0.5, 1.0, 'Total Hybrid jobs per industry')