In [39]:
```
import pandas as pd
excel_file_path = 'Excel test.xlsx'
df = pd.read_excel(excel_file_path, sheet_name='Question 1 data')
df
```

Out[39]:

	Age	Number of People
0	0	23
1	1	36
2	2	49
3	3	56
4	4	90
5	5	47
6	6	89
7	7	55
8	8	47
9	9	74
10	10	45
11	11	77
12	12	73
13	13	50
14	14	36
15	15	28
16	16	47
17	17	47
18	18	53
19	19	77
20	20	88
21	21	74
22	22	65
23	23	37
24	24	53
25	25	74
26	26	23
27	27	43
28	28	52
29	29	49
30	30	25

In [40]:
```
import numpy as np
weighted_Average_age = np.average(df['Age'], weights=df['Number of People'])
weighted_Average_age
```

Out[40]:
```
14.670630202140309
```

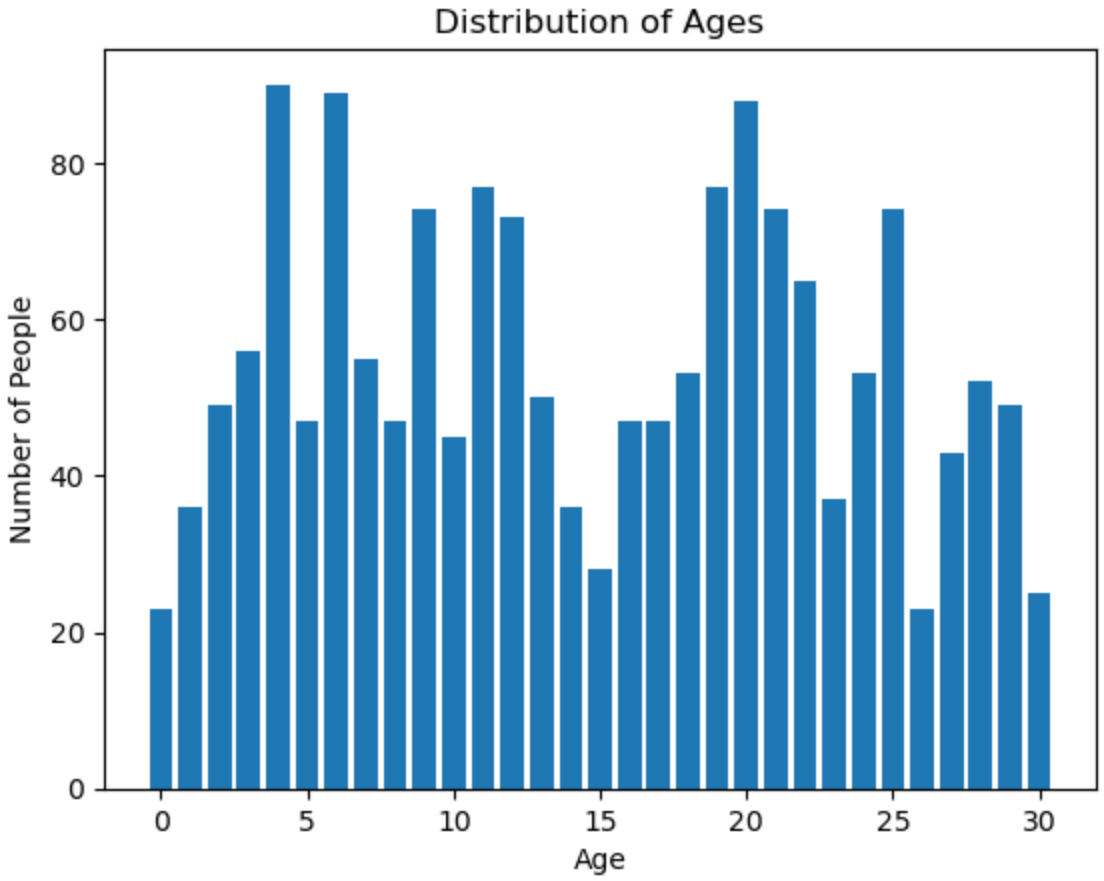
In [41]:
```
weighted_median_age = np.median(np.repeat(df['Age'], df['Number of People']))
weighted_median_age
```

Out[41]:
```
14.0
```

In [8]:
```
# discrepancy might be due to the distribution of ages in your data. The average is
#influenced by the weights (number of people in each age group), and the median is aff
```

In [42]:
```
import matplotlib.pyplot as plt

# Create a bar chart
plt.bar(df['Age'], df['Number of People'])
plt.xlabel('Age')
plt.ylabel('Number of People')
plt.title('Distribution of Ages')
plt.show()
```



In [43]:
```
df2 = pd.read_excel(excel_file_path, sheet_name='Question 2 data')
df2.head()
```

Out[43]:

	ID	Profession	Unnamed: 2	ID.1	City	Unnamed: 5	ID.2	Years of Experience
0	42934	Physician	NaN	42934.0	Scarborough	NaN	11173.0	7.0
1	17024	Physician	NaN	17024.0	Guelph	NaN	12440.0	3.0
2	16941	Physician	NaN	16941.0	Toronto	NaN	12955.0	10.0
3	28056	Physician	NaN	28843.0	Toronto	NaN	16941.0	9.0
4	28843	Health Care Administrator	NaN	87069.0	Thunder Bay	NaN	17024.0	10.0

In [44]:
```
df_profession=df2.iloc[:, :2]
df_profession.head()
```

Out[44]:

	ID	Profession
0	42934	Physician
1	17024	Physician
2	16941	Physician
3	28056	Physician
4	28843	Health Care Administrator

In [45]:
```
df_city=df2.iloc[:, 3:5]
df_city.head()
```

Out[45]:

	ID.1	City
0	42934.0	Scarborough
1	17024.0	Guelph
2	16941.0	Toronto
3	28843.0	Toronto
4	87069.0	Thunder Bay

In [46]:
```
df_year_exp=df2.iloc[:, 6:9]
df_year_exp.head()
```

Out[46]:

	ID.2	Years of Experience
0	11173.0	7.0
1	12440.0	3.0
2	12955.0	10.0
3	16941.0	9.0
4	17024.0	10.0

In [47]:
```
merged_df1 = pd.merge(df_profession, df_city, left_on='ID', right_on='ID.1')
merged_df1.head()
```

Out[47]:

	ID	Profession	ID.1	City
0	42934	Physician	42934.0	Scarborough
1	17024	Physician	17024.0	Guelph
2	16941	Physician	16941.0	Toronto
3	28843	Health Care Administrator	28843.0	Toronto
4	87069	Allied Health Professional	87069.0	Thunder Bay

In [48]:
```
# Merge the result with df_year_exp on 'ID.2'
final_merged_df = pd.merge(merged_df1, df_year_exp, left_on='ID', right_on='ID.2')
final_merged_df.head()
```

Out[48]:

	ID	Profession	ID.1	City	ID.2	Years of Experience
0	17024	Physician	17024.0	Guelph	17024.0	10.0
1	16941	Physician	16941.0	Toronto	16941.0	9.0
2	28843	Health Care Administrator	28843.0	Toronto	28843.0	12.0
3	87069	Allied Health Professional	87069.0	Thunder Bay	87069.0	4.0
4	34501	Physician	34501.0	Toronto	34501.0	8.0

In [49]:
```
# Drop redundant columns
final_merged_df = final_merged_df.drop(['ID.1', 'ID.2'], axis=1)
final_merged_df.head()
```

Out[49]:

	ID	Profession	City	Years of Experience
0	17024	Physician	Guelph	10.0
1	16941	Physician	Toronto	9.0
2	28843	Health Care Administrator	Toronto	12.0
3	87069	Allied Health Professional	Thunder Bay	4.0
4	34501	Physician	Toronto	8.0

In [55]:
```
final_merged_df['Profession '].value_counts()
```

Out[55]:
```
Physician                    15
Health Care Administrator     7
Allied Health Professional    6
Nurse                         5
Name: Profession , dtype: int64
```

In [59]:
```
To=final_merged_df[final_merged_df['City']=='Toronto']
To['Profession '].value_counts()
```

Out[59]:
```
Physician                     5
Health Care Administrator     2
Allied Health Professional    2
Nurse                         1
Name: Profession , dtype: int64
```

In [63]:
```
To['Years of Experience'].sum()
```

Out[63]:
```
85.0
```

In [67]:
```
final_merged_df[final_merged_df['Profession ']!= 'Physician'].mean()
```

```
C:\Users\rozap\AppData\Local\Temp\ipykernel_15096\2564553430.py:1: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depre
cated; in a future version this will raise TypeError.  Select only valid columns befo
re calling the reduction.
  final_merged_df[final_merged_df['Profession ']!= 'Physician'].mean()
```

Out[67]:
```
ID                     34553.388889
Years of Experience        6.722222
dtype: float64
```

In [70]:
```
df3 = pd.read_excel(excel_file_path, sheet_name='Question 3 (2)')
df3
```

Out[70]:

	Physician Name	Earnings from TM Premium	Earnings From TM Visits	Overall Earnings
0	Physician97178	14140	40026.90	437919.22
1	Physician949793	30830	96183.25	692876.51
2	Physician948003	57090	150391.35	689164.33
3	Physician93592	19195	91026.05	395235.62
4	Physician915089	52060	149852.75	1002942.26
...	...	...	...	...
85	Physician246349	11455	44605.31	253282.74
86	Physician244077	28075	79143.90	606808.42
87	Physician243013	42000	143889.60	515567.66
88	Physician209125	43095	127801.85	580204.23
89	Physician116056	15360	49363.15	609113.67

90 rows × 4 columns

In [77]:
```
# defining new column to calculate proportion
df3['Proportion TM'] = df3['Earnings from TM Premium'] / df3['Overall Earnings']

# Set a threshold for low proportion
threshold = 0.03

# Identify physicians with a low proportion of telemedicine earnings
low_risk_physicians = df3[df3['Proportion TM'] < threshold]

#printing the names of physicians
low_risk_physicians
```

Out[77]:

	Physician Name	Earnings from TM Premium	Earnings From TM Visits	Overall Earnings	Proportion TM
5	Physician91385	21805	70229.60	840244.59	0.025951
9	Physician84777	12170	108525.05	466099.37	0.026110
10	Physician839680	26750	80029.70	1079291.74	0.024785
12	Physician835981	10790	33058.00	604850.37	0.017839
18	Physician79272	16580	53034.55	602001.20	0.027541
26	Physician75002	11945	46750.16	502779.13	0.023758
29	Physician73442	12620	34616.25	588440.78	0.021447
37	Physician598457	35445	104660.15	1208486.06	0.029330
38	Physician590626	16045	60087.10	623044.61	0.025753
41	Physician55449	15360	48460.25	973088.09	0.015785
45	Physician517788	17660	52135.10	1007922.64	0.017521
54	Physician419845	17035	61058.20	841373.53	0.020247
62	Physician417112	10285	25549.40	350707.34	0.029326
66	Physician414906	15910	78924.20	750234.74	0.021207
71	Physician412314	15720	48565.85	702623.90	0.022373
72	Physician412291	15690	62172.40	576319.95	0.027224
76	Physician411353	21105	65319.40	732672.13	0.028806
78	Physician410464	10720	27120.80	784742.71	0.013661
81	Physician352164	12870	38524.65	547436.16	0.023510
82	Physician352086	11715	36120.70	492104.11	0.023806
89	Physician116056	15360	49363.15	609113.67	0.025217