In [1]:

```
import pandas as pd
pd.set_option('Display.max_columns',500)
import warnings
warnings.filterwarnings(action='ignore')
```

In [2]:

```
df = pd.read_csv('survey_results_public.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Stude
0	1	I am a student who is learning to code	Yes	Never	The quality of OSS and closed source software ...	Not employed, and not looking for work	United Kingdom	N
1	2	I am a student who is learning to code	No	Less than once per year	The quality of OSS and closed source software ...	Not employed, but looking for work	Bosnia and Herzegovina	Yes, fu tir
2	3	I am not primarily a developer, but I write co...	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	Thailand	N
3	4	I am a developer by profession	No	Never	The quality of OSS and closed source software ...	Employed full-time	United States	N
4	5	I am a developer by profession	Yes	Once a month or more often	OSS is, on average, of HIGHER quality than pro...	Employed full-time	Ukraine	N

In [4]:

```
#lets drop some useless columns
df.columns
```

Out[4]:

```
Index(['Respondent', 'MainBranch', 'Hobbyist', 'OpenSourcer', 'OpenSource',
       'Employment', 'Country', 'Student', 'EdLevel', 'UndergradMajor',
       'EduOther', 'OrgSize', 'DevType', 'YearsCode', 'Age1stCode',
       'YearsCodePro', 'CareerSat', 'JobSat', 'MgrIdiot', 'MgrMoney',
       'MgrWant', 'JobSeek', 'LastHireDate', 'LastInt', 'FizzBuzz',
       'JobFactors', 'ResumeUpdate', 'CurrencySymbol', 'CurrencyDesc',
       'CompTotal', 'CompFreq', 'ConvertedComp', 'WorkWeekHrs', 'WorkPlan',
       'WorkChallenge', 'WorkRemote', 'WorkLoc', 'ImpSyn', 'CodeRev',
       'CodeRevHrs', 'UnitTests', 'PurchaseHow', 'PurchaseWhat',
       'LanguageWorkedWith', 'LanguageDesireNextYear', 'DatabaseWorkedWith',
       'DatabaseDesireNextYear', 'PlatformWorkedWith',
       'PlatformDesireNextYear', 'WebFrameWorkedWith',
       'WebFrameDesireNextYear', 'MiscTechWorkedWith',
       'MiscTechDesireNextYear', 'DevEnviron', 'OpSys', 'Containers',
       'BlockchainOrg', 'BlockchainIs', 'BetterLife', 'ITperson', 'OffOn',
       'SocialMedia', 'Extraversion', 'ScreenName', 'SOVisit1st',
       'SOVisitFreq', 'SOVisitTo', 'SOFindAnswer', 'SOTimeSaved',
       'SOHowMuchTime', 'SOAccount', 'SOPartFreq', 'SOJobs', 'EntTeams',
       'SOComm', 'WelcomeChange', 'SONewContent', 'Age', 'Gender', 'Trans',
       'Sexuality', 'Ethnicity', 'Dependents', 'SurveyLength', 'SurveyEase'],
      dtype='object')
```

In [5]:

```
df1 = df.drop(['Respondent','OpenSourcer', 'OpenSource','CareerSat', 'JobSat', 'MgrI
        'MgrWant','LastInt', 'FizzBuzz','JobFactors', 'ResumeUpdate','WorkChallenge',
        'CodeRevHrs', 'UnitTests', 'PurchaseHow', 'PurchaseWhat','PlatformWorkedWith'
        'PlatformDesireNextYear','Containers',
        'BlockchainOrg', 'BlockchainIs','Extraversion', 'ScreenName','SOPartFreq', 'S
        'SOComm','WelcomeChange', 'SONewContent'],axis=1)
```

In [6]:

```
df1.head()
```

Out[6]:

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
0	I am a student who is learning to code	Yes	Not employed, and not looking for work	United Kingdom	No	Primary/elementary school	NaN
1	I am a student who is learning to code	No	Not employed, but looking for work	Bosnia and Herzegovina	Yes, full-time	Secondary school (e.g. American high school, G...	NaN
2	I am not primarily a developer, but I write co...	Yes	Employed full-time	Thailand	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Web development or web design
3	I am a developer by profession	No	Employed full-time	United States	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
4	I am a developer by profession	Yes	Employed full-time	Ukraine	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

In [7]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88883 entries, 0 to 88882
Data columns (total 52 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   MainBranch              88331 non-null  object 
 1   Hobbyist                88883 non-null  object 
 2   Employment              87181 non-null  object 
 3   Country                 88751 non-null  object 
 4   Student                 87014 non-null  object 
 5   EdLevel                 86390 non-null  object 
 6   UndergradMajor          75614 non-null  object 
 7   EduOther                84260 non-null  object 
 8   OrgSize                 71791 non-null  object 
 9   DevType                 81335 non-null  object 
 10  YearsCode               87938 non-null  object 
 11  Age1stCode              87634 non-null  object 
 12  YearsCodePro            74331 non-null  object 
 13  JobSeek                 80555 non-null  object 
 14  LastHireDate            79854 non-null  object 
 15  CurrencySymbol          71392 non-null  object 
 16  CurrencyDesc            71392 non-null  object 
 17  CompTotal               55945 non-null  float64
 18  CompFreq                63268 non-null  object 
 19  ConvertedComp           55823 non-null  float64
 20  WorkWeekHrs             64503 non-null  float64
 21  WorkPlan                68914 non-null  object 
 22  WorkLoc                 70055 non-null  object 
 23  LanguageWorkedWith      87569 non-null  object 
 24  LanguageDesireNextYear  84088 non-null  object 
 25  DatabaseWorkedWith      76026 non-null  object 
 26  DatabaseDesireNextYear  69147 non-null  object 
 27  WebFrameWorkedWith      65022 non-null  object 
 28  WebFrameDesireNextYear  62944 non-null  object 
 29  MiscTechWorkedWith      59586 non-null  object 
 30  MiscTechDesireNextYear  64511 non-null  object 
 31  DevEnviron              87317 non-null  object 
 32  OpSys                   87851 non-null  object 
 33  BetterLife              86269 non-null  object 
 34  ITperson                87141 non-null  object 
 35  OffOn                   86663 non-null  object 
 36  SocialMedia             84437 non-null  object 
 37  SOVisit1st              83877 non-null  object 
 38  SOVisitFreq             88263 non-null  object 
 39  SOVisitTo               88086 non-null  object 
 40  SOFindAnswer            87816 non-null  object 
 41  SOTimeSaved             86344 non-null  object 
 42  SOHowMuchTime           68378 non-null  object 
 43  SOAccount               87828 non-null  object 
 44  Age                     79210 non-null  float64
 45  Gender                  85406 non-null  object 
 46  Trans                   83607 non-null  object 
 47  Sexuality               76147 non-null  object 
 48  Ethnicity               76668 non-null  object 
 49  Dependents              83059 non-null  object 
 50  SurveyLength            86984 non-null  object 
 51  SurveyEase              87081 non-null  object 
dtypes: float64(4), object(48)
memory usage: 35.3+ MB
```

In [8]:

```
# will check null values

for null in df1.columns:
    print("Missing values count in {} is {} and {}% ".format(null,df1[null].isnull()
```

```
Missing values count in MainBranch is 552 and 0.006210411439757884%
Missing values count in Hobbyist is 0 and 0.0%
Missing values count in Employment is 1702 and 0.01914876860592014%
Missing values count in Country is 132 and 0.0014850983877681895%
Missing values count in Student is 1869 and 0.02102764308135414%
Missing values count in EdLevel is 2493 and 0.0280481081871674%
Missing values count in UndergradMajor is 13269 and 0.1492861402067887%
Missing values count in EduOther is 4623 and 0.05201219580797228%
Missing values count in OrgSize is 17092 and 0.1922977397252568%
Missing values count in DevType is 7548 and 0.08492062599147193%
Missing values count in YearsCode is 945 and 0.010631954366976813%
Missing values count in Age1stCode is 1249 and 0.014052180956988401%
Missing values count in YearsCodePro is 14552 and 0.16372084650608104%
Missing values count in JobSeek is 8328 and 0.09369620737373852%
Missing values count in LastHireDate is 9029 and 0.10158297987241655%
Missing values count in CurrencySymbol is 17491 and 0.196786787124647%
Missing values count in CurrencyDesc is 17491 and 0.196786787124647%
Missing values count in CompTotal is 32938 and 0.37057705072961084%
Missing values count in CompFreq is 25615 and 0.28818784244456197%
Missing values count in ConvertedComp is 33060 and 0.3719496416637602%
Missing values count in WorkWeekHrs is 24380 and 0.27429317192263986%
Missing values count in WorkPlan is 19969 and 0.2246661341313862%
Missing values count in WorkLoc is 18828 and 0.21182903367348085%
Missing values count in LanguageWorkedWith is 1314 and 0.014783479405510615%
Missing values count in LanguageDesireNextYear is 4795 and 0.05394732401021567%
Missing values count in DatabaseWorkedWith is 12857 and 0.14465083311769406%
Missing values count in DatabaseDesireNextYear is 19736 and 0.2220447104620681%
Missing values count in WebFrameWorkedWith is 23861 and 0.268454035079824%
Missing values count in WebFrameDesireNextYear is 25939 and 0.2918330839418111%
Missing values count in MiscTechWorkedWith is 29297 and 0.3296130868670049%
Missing values count in MiscTechDesireNextYear is 24372 and 0.2742031659597448%
Missing values count in DevEnviron is 1566 and 0.01761866723670443%
Missing values count in OpSys is 1032 and 0.011610769213460392%
Missing values count in BetterLife is 2614 and 0.029409448375954907%
Missing values count in ITperson is 1742 and 0.019598798420395535%
Missing values count in OffOn is 2220 and 0.02497665470337741%
Missing values count in SocialMedia is 4446 and 0.050020813878919476%
Missing values count in SOVisit1st is 5006 and 0.05632123128157240 4%
Missing values count in SOVisitFreq is 620 and 0.006975462124365739%
Missing values count in SOVisitTo is 797 and 0.008966844053418539%
Missing values count in SOFindAnswer is 1067 and 0.012004545301126199%
Missing values count in SOTimeSaved is 2539 and 0.02856564247381389%
Missing values count in SOHowMuchTime is 20505 and 0.230696533645354%
Missing values count in SOAccount is 1055 and 0.011869536356783637%
Missing values count in Age is 9673 and 0.10882845988546741%
Missing values count in Gender is 3477 and 0.039118841623257544%
Missing values count in Trans is 5276 and 0.05935893252928007%
Missing values count in Sexuality is 12736 and 0.14328949292890655%
Missing values count in Ethnicity is 12215 and 0.13742785459536694%
Missing values count in Dependents is 5824 and 0.06552434098759043%
Missing values count in SurveyLength is 1899 and 0.021365165442210548%
Missing values count in SurveyEase is 1802 and 0.020273843142108165%
```

Almost every column has missing values. Lets fix them.

In [9]:

```
df1['OrgSize'].unique()
```

Out[9]:

```
array([nan, '100 to 499 employees', '10,000 or more employees',
       'Just me - I am a freelancer, sole proprietor, etc.',
       '10 to 19 employees', '20 to 99 employees',
       '1,000 to 4,999 employees', '2-9 employees',
       '500 to 999 employees', '5,000 to 9,999 employees'], dtype=object)
```

In [10]:

```
df1['OrgSize'] = df1['OrgSize'].map({'20 to 99 employees':70,'100 to 499 employee':3
                   '1,000 to 4,999 employees':4500,'2-9 employees': 9,'10 to 19 empl
                   '500 to 999 employees':800,'Just me - I am a freelancer, sole pro
                   '5,000 to 9,999 employees':9999,'nan':0})
```

In [11]:

```
meadian = df1['OrgSize'].median()
```

In [12]:

```
df1['OrgSize'].fillna(meadian,inplace=True)
```

In [13]:

```
df1['YearsCode'].unique()
```

Out[13]:

```
array(['4', nan, '3', '16', '13', '6', '8', '12', '2', '5', '17', '10',
       '14', '35', '7', 'Less than 1 year', '30', '9', '26', '40', '19',
       '15', '20', '28', '25', '1', '22', '11', '33', '50', '41', '18',
       '34', '24', '23', '42', '27', '21', '36', '32', '39', '38', '31',
       '37', 'More than 50 years', '29', '44', '45', '48', '46', '43',
       '47', '49'], dtype=object)
```

In [18]:

```
df1['YearsCode'].replace('Less than 1 year',0,inplace=True)
```

In [19]:

```
df1['YearsCode'].replace('More than 50 years',51,inplace=True)
```

In [23]:

```
df1['YearsCode'].median()
```

Out[23]:

```
9.0
```

In [ ]:

In [24]:

```
df1['YearsCode'].fillna(9,inplace=True)
```

In [26]:

```
df1['YearsCode'].unique()
```

Out[26]:

```
array(['4', 9, '3', '16', '13', '6', '8', '12', '2', '5', '17', '10',
       '14', '35', '7', 0, '30', '9', '26', '40', '19', '15', '20', '28',
       '25', '1', '22', '11', '33', '50', '41', '18', '34', '24', '23',
       '42', '27', '21', '36', '32', '39', '38', '31', '37', 51, '29',
       '44', '45', '48', '46', '43', '47', '49'], dtype=object)
```

In [27]:

```
df1['Age1stCode'].replace('Younger than 5 years',5,inplace=True)
df1['Age1stCode'].replace('Older than 85',85,inplace=True)
```

In [31]:

```
df1['Age1stCode'].median()
```

Out[31]:

```
15.0
```

In [32]:

```
df1['Age1stCode'].fillna(15,inplace=True)
```

In [34]:

```
df1['Age1stCode'].unique()
```

Out[34]:

```
array(['10', '17', '22', '16', '14', '15', '11', '20', '13', '18', '12',
       '19', '21', '8', '35', '6', '9', '29', '7', '5', '23', '30', 15,
       '27', '24', 5, '33', '25', '26', '39', '36', '38', '28', '31', 85,
       '32', '37', '50', '65', '42', '34', '40', '67', '43', '44', '60',
       '46', '45', '49', '51', '41', '55', '83', '48', '53', '54', '47',
       '56', '79', '61', '68', '77', '66', '52', '80', '62', '84', '57',
       '58', '63'], dtype=object)
```

In [35]:

```
df1['YearsCodePro'].replace('Less than 1 year',0,inplace=True)
df1['YearsCodePro'].replace('More than 50 years',51,inplace=True)
```

In [37]:

```
df1['YearsCodePro'].median()
```

Out[37]:

```
6.0
```

In [38]:

```
df1['YearsCodePro'].fillna(6,inplace=True)
```

In [39]:

```
df1['OrgSize'] = df1['OrgSize'].astype(int)
df1['YearsCode'] = df1['YearsCode'].astype(int)
df1['Age1stCode'] = df1['Age1stCode'].astype(int)
df1['YearsCodePro'] = df1['YearsCodePro'].astype(int)
```

In [44]:

```
# as final salary amunt is in US dollar so dropping related features
df1.drop(['LastHireDate','CurrencySymbol','CurrencyDesc','CompTotal','CompFreq'],axi
```

In [45]:

```
df1.head()
```

Out[45]:

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
0	I am a student who is learning to code	Yes	Not employed, and not looking for work	United Kingdom	No	Primary/elementary school	NaN
1	I am a student who is learning to code	No	Not employed, but looking for work	Bosnia and Herzegovina	Yes, full-time	Secondary school (e.g. American high school, G...	NaN
2	I am not primarily a developer, but I write co...	Yes	Employed full-time	Thailand	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Web development or web design

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
3	I am a developer by profession	No	Employed full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
4	I am a developer by profession	Yes	Employed full-time	Ukraine	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

In [49]:

```
df1['ConvertedComp'].median()
```

Out[49]:

```
57287.0
```

In [50]:

```
df1['ConvertedComp'].fillna(57287.0,inplace=True)
```

In [54]:

```
df1['WorkWeekHrs'].median()
```

Out[54]:

```
40.0
```

In [55]:

```
df1['WorkWeekHrs'].fillna(40,inplace=True)
```

In [58]:

```
df1.drop(['WebFrameWorkedWith','WebFrameDesireNextYear','MiscTechWorkedWith','MiscTe
         'Trans','Sexuality','Dependents'],axis=1,inplace=True)
```

In [59]:

```
df1.head()
```

Out[59]:

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
0	I am a student who is learning to code	Yes	Not employed, and not looking for work	United Kingdom	No	Primary/elementary school	NaN
1	I am a student who is learning to code	No	Not employed, but looking for work	Bosnia and Herzegovina	Yes, full-time	Secondary school (e.g. American high school, G...	NaN
2	I am not primarily a developer, but I write co...	Yes	Employed full-time	Thailand	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Web development or web design

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
3	I am a developer by profession	No	Employed full-time	United States	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
4	I am a developer by profession	Yes	Employed full-time	Ukraine	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

In [60]:

```
df1['SOVisit1st'].unique()
```

Out[60]:

```
array(['2017', '2011', '2014', "I don't remember", '2012', '2013', nan,
       '2010', '2018', '2008', '2016', '2015', '2009', '2019'],
      dtype=object)
```

In [63]:

```
df1['SOVisit1st'].replace("I don't remember",2008,inplace=True)
```

In [64]:

```
df1['SOVisit1st'].mode()
```

Out[64]:

```
0    2008
dtype: object
```

In [65]:

```
df1['SOVisit1st'].fillna(2008,inplace=True)
```

In [67]:

```
df1['SOVisit1st'] = df1['SOVisit1st'].astype(int)
```

In [68]:

```
df1['SOFindAnswer'].unique()
```

Out[68]:

```
array(['3-5 times per week', '6-10 times per week', '1-2 times per week',
       'More than 10 times per week', 'Less than once per week', nan],
      dtype=object)
```

In [84]:

```
df1['SOFindAnswer'].mode()
```

Out[84]:

```
0    1-2 times per week
dtype: object
```

In [85]:

```
df1['SOFindAnswer'].fillna('1-2 times per week',inplace=True)
```

In [86]:

```
df1['SOFindAnswer'].unique()
```

Out[86]:

```
array(['3-5 times per week', '6-10 times per week', '1-2 times per week',
       'More than 10 times per week', 'Less than once per week'],
      dtype=object)
```

In [88]:

```
df1['visit_per_week'] = df1['SOFindAnswer'].map({'3-5 times per week':4, '6-10 times
                                                  'More than 10 times per week':12, 'Less than once per week':
```

In [90]:
```
df1['SOHowMuchTime'].unique()
```

Out[90]:
```
array(['31-60 minutes', '11-30 minutes', nan, '60+ minutes',
       '0-10 minutes'], dtype=object)
```

In [91]:
```
df1['SOHowMuchTime'].mode()
```

Out[91]:
```
0    11-30 minutes
dtype: object
```

In [92]:
```
df1['SOHowMuchTime'].fillna('11-30 minutes',inplace=True)
```

In [95]:
```
df1['Mintues_spend'] = df1['SOHowMuchTime'].map({'31-60 minutes':60,'11-30 minutes':
```

In [96]:
```
df1['Age'].unique()
```

Out[96]:
```
array([14. , 19. , 28. , 22. , 30. , 42. , 24. , 23. ,  nan, 21. , 31. ,
       20. , 26. , 29. , 38. , 47. , 34. , 32. , 25. , 17. , 35. , 27. ,
       44. , 43. , 62. , 37. , 45. , 18. , 33. , 36. , 16. , 39. , 64. ,
       41. , 54. , 49. , 40. , 56. , 12. , 58. , 46. , 59. , 51. , 48. ,
       57. , 52. , 50. , 23.9, 55. , 15. , 67. , 13. ,  1. , 53. , 69. ,
       65. , 17.5, 63. , 61. , 68. , 73. , 70. , 60. , 16.5, 46.5, 11. ,
       71. ,  3. , 97. , 29.5, 77. , 74. , 26.5, 26.3, 24.5, 78. , 72. ,
       66. , 76. , 10. , 75. , 99. , 83. , 79. , 36.8, 14.1, 13.5, 19.5,
       98. , 43.5, 22.5, 31.5, 21.5, 28.5, 33.6,  2. , 38.5, 30.8, 24.8,
       90. , 61.3, 81. ,  4. , 17.3, 19.9, 80. , 85. , 88. , 23.5, 16.9,
       20.9, 91. , 98.9, 57.9,  9. , 94. , 95. , 37.5, 14.5,  5. , 82. ,
       84. , 37.3, 33.5, 53.8, 31.4, 87. ])
```

In [98]:
```
df1['Age'].mean()
```

Out[98]:
```
30.336698649160446
```

In [99]:
```
df1['Age'].fillna(29,inplace=True)
```

In [100…
```
df1['Gender'].unique()
```

Out[100…
```
array(['Man', nan, 'Woman',
       'Non-binary, genderqueer, or gender non-conforming',
       'Woman;Non-binary, genderqueer, or gender non-conforming',
       'Woman;Man;Non-binary, genderqueer, or gender non-conforming',
       'Woman;Man',
       'Man;Non-binary, genderqueer, or gender non-conforming'],
      dtype=object)
```

In [102…
```
# Gender feature not giving much info. will drop it
df1.head()
```

Out[102…

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
0	I am a student who is learning to code	Yes	Not employed, and not looking for work	United Kingdom	No	Primary/elementary school	NaN
1	I am a student who is learning to code	No	Not employed, but looking for work	Bosnia and Herzegovina	Yes, full-time	Secondary school (e.g. American high school, G...	NaN
2	I am not primarily a developer, but I write co...	Yes	Employed full-time	Thailand	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Web development or web design
3	I am a developer by profession	No	Employed full-time	United States	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
4	I am a developer by profession	Yes	Employed full-time	Ukraine	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

In [103…

```
df1.drop(['SOFindAnswer','SOHowMuchTime','Gender','Mintues_'],axis=1,inplace=True)
```

In [105…

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88883 entries, 0 to 88882
Data columns (total 37 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   MainBranch            88331 non-null  object 
 1   Hobbyist              88883 non-null  object 
 2   Employment            87181 non-null  object 
 3   Country               88751 non-null  object 
 4   Student               87014 non-null  object 
 5   EdLevel               86390 non-null  object 
 6   UndergradMajor        75614 non-null  object 
 7   EduOther              84260 non-null  object 
 8   OrgSize               88883 non-null  int32  
 9   DevType               81335 non-null  object 
 10  YearsCode             88883 non-null  int32  
 11  Age1stCode            88883 non-null  int32  
 12  YearsCodePro          88883 non-null  int32  
 13  JobSeek               80555 non-null  object 
 14  ConvertedComp         88883 non-null  float64
 15  WorkWeekHrs           88883 non-null  float64
 16  WorkPlan              68914 non-null  object 
 17  WorkLoc                 70055 non-null  object 
 18  LanguageWorkedWith      87569 non-null  object 
 19  LanguageDesireNextYear  84088 non-null  object 
 20  DatabaseWorkedWith      76026 non-null  object 
 21  DatabaseDesireNextYear  69147 non-null  object 
 22  DevEnviron              87317 non-null  object 
 23  OpSys                   87851 non-null  object 
 24  BetterLife              86269 non-null  object 
 25  OffOn                   86663 non-null  object 
 26  SocialMedia             84437 non-null  object 
 27  SOVisit1st              88883 non-null  int32  
 28  SOVisitFreq             88263 non-null  object 
 29  SOVisitTo               88086 non-null  object 
 30  SOTimeSaved             86344 non-null  object 
 31  SOAccount               87828 non-null  object 
 32  Age                     88883 non-null  float64
 33  SurveyLength            86984 non-null  object 
 34  SurveyEase              87081 non-null  object 
 35  visit_per_week          88883 non-null  int64  
 36  Mintues_spend           88883 non-null  int64  
dtypes: float64(3), int32(5), int64(2), object(27)
memory usage: 23.4+ MB
```

In [126…

```python
catogerical_feature = df1.select_dtypes(object)
```

In [128…

```python
for null in catogerical_feature:
    df2[null].fillna(df2[null].mode()[0],inplace=True)
```

In [130…

```python
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88883 entries, 0 to 88882
Data columns (total 37 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   MainBranch              88883 non-null  object 
 1   Hobbyist                88883 non-null  object 
 2   Employment              88883 non-null  object 
 3   Country                 88883 non-null  object 
 4   Student                 88883 non-null  object 
 5   EdLevel                 88883 non-null  object 
 6   UndergradMajor          88883 non-null  object 
 7   EduOther                88883 non-null  object 
 8   OrgSize                 88883 non-null  int32  
 9   DevType                 88883 non-null  object 
 10  YearsCode               88883 non-null  int32  
 11  Age1stCode              88883 non-null  int32  
 12  YearsCodePro            88883 non-null  int32  
 13  JobSeek                 88883 non-null  object 
 14  ConvertedComp           88883 non-null  float64
 15  WorkWeekHrs             88883 non-null  float64
 16  WorkPlan                88883 non-null  object 
 17  WorkLoc                 88883 non-null  object 
 18  LanguageWorkedWith      88883 non-null  object 
 19  LanguageDesireNextYear  88883 non-null  object 
 20  DatabaseWorkedWith      88883 non-null  object 
 21  DatabaseDesireNextYear  88883 non-null  object 
 22  DevEnviron              88883 non-null  object 
 23  OpSys                   88883 non-null  object 
 24  BetterLife              88883 non-null  object 
 25  OffOn                   88883 non-null  object 
 26  SocialMedia             88883 non-null  object
 27  SOVisit1st              88883 non-null  int32
 28  SOVisitFreq             88883 non-null  object
 29  SOVisitTo               88883 non-null  object
 30  SOTimeSaved             88883 non-null  object
 31  SOAccount               88883 non-null  object
 32  Age                     88883 non-null  float64
 33  SurveyLength            88883 non-null  object
 34  SurveyEase              88883 non-null  object
 35  visit_per_week          88883 non-null  int64
 36  Mintues_spend           88883 non-null  int64
dtypes: float64(3), int32(5), int64(2), object(27)
memory usage: 23.4+ MB
```

In [131…

```
# Now we do not have any null values.
df2.head()
```

Out[131…

	MainBranch	Hobbyist	Employment	Country	Student	EdLevel	UndergradMajor
0	I am a student who is learning to code	Yes	Not employed, and not looking for work	United Kingdom	No	Primary/elementary school	Computer science, computer engineering, or sof...
1	I am a student who is learning to code	No	Not employed, but looking for work	Bosnia and Herzegovina	Yes, full-time	Secondary school (e.g. American high school, G...	Computer science, computer engineering, or sof...
2	I am not primarily a developer, but I write co...	Yes	Employed full-time	Thailand	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Web development or web design
3	I am a developer by profession	No	Employed full-time	United States	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
4	I am a developer by profession	Yes	Employed full-time	Ukraine	No	Bachelor’s degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

In [135…

```
df2['MainBranch'].value_counts()
```

Out[135…

```
I am a developer by profession                                                   662
31
I am a student who is learning to code                                           101
89
I am not primarily a developer, but I write code sometimes as part of my work     75
39
I code primarily as a hobby                                                       33
```

```
40
I used to be a developer by profession, but no longer am                    15
84
Name: MainBranch, dtype: int64
```

In [136…

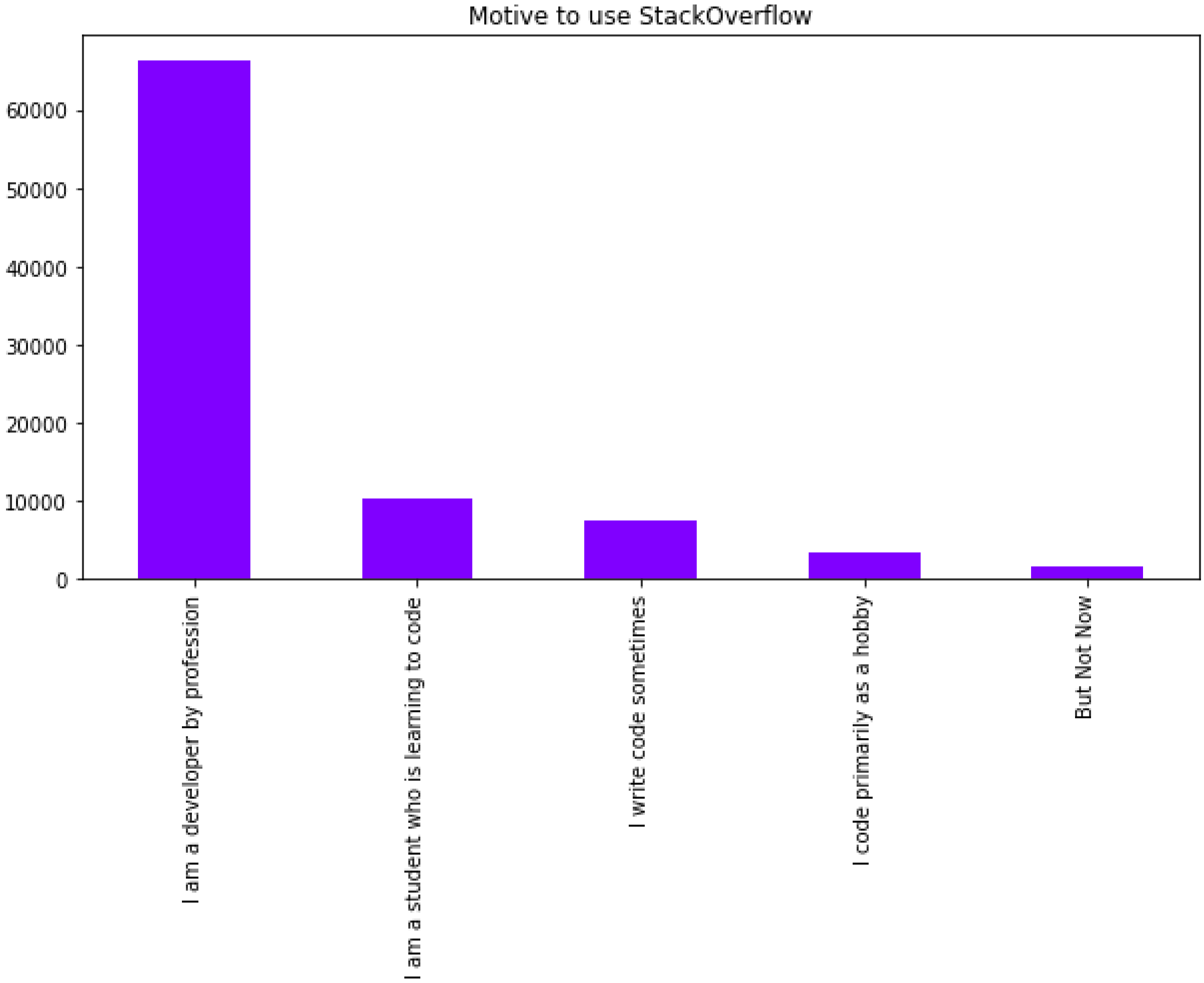
```
df2['MainBranch'].replace('I am not primarily a developer, but I write code sometime
df2['MainBranch'].replace('I used to be a developer by profession, but no longer am'
```

In [153…

```
df2['MainBranch'].value_counts().plot(kind='bar',figsize=(10,5),colormap='rainbow')
plt.title('Motive to use StackOverflow')
```

Out[153…

```
Text(0.5, 1.0, 'Motive to use StackOverflow')
```



We can see that most of the users are professional developers. Students also use it for study

In [139…

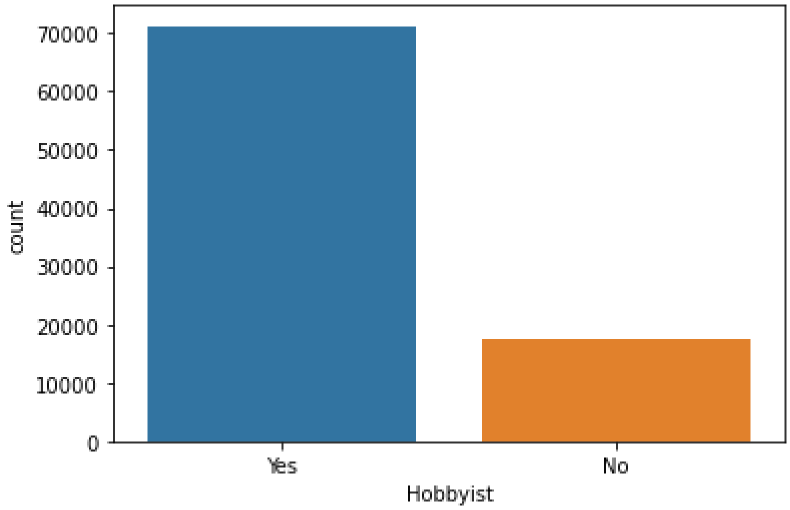
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [159…

```
sns.countplot(df2['Hobbyist'])
```

Out[159…

```
<AxesSubplot:xlabel='Hobbyist', ylabel='count'>
```

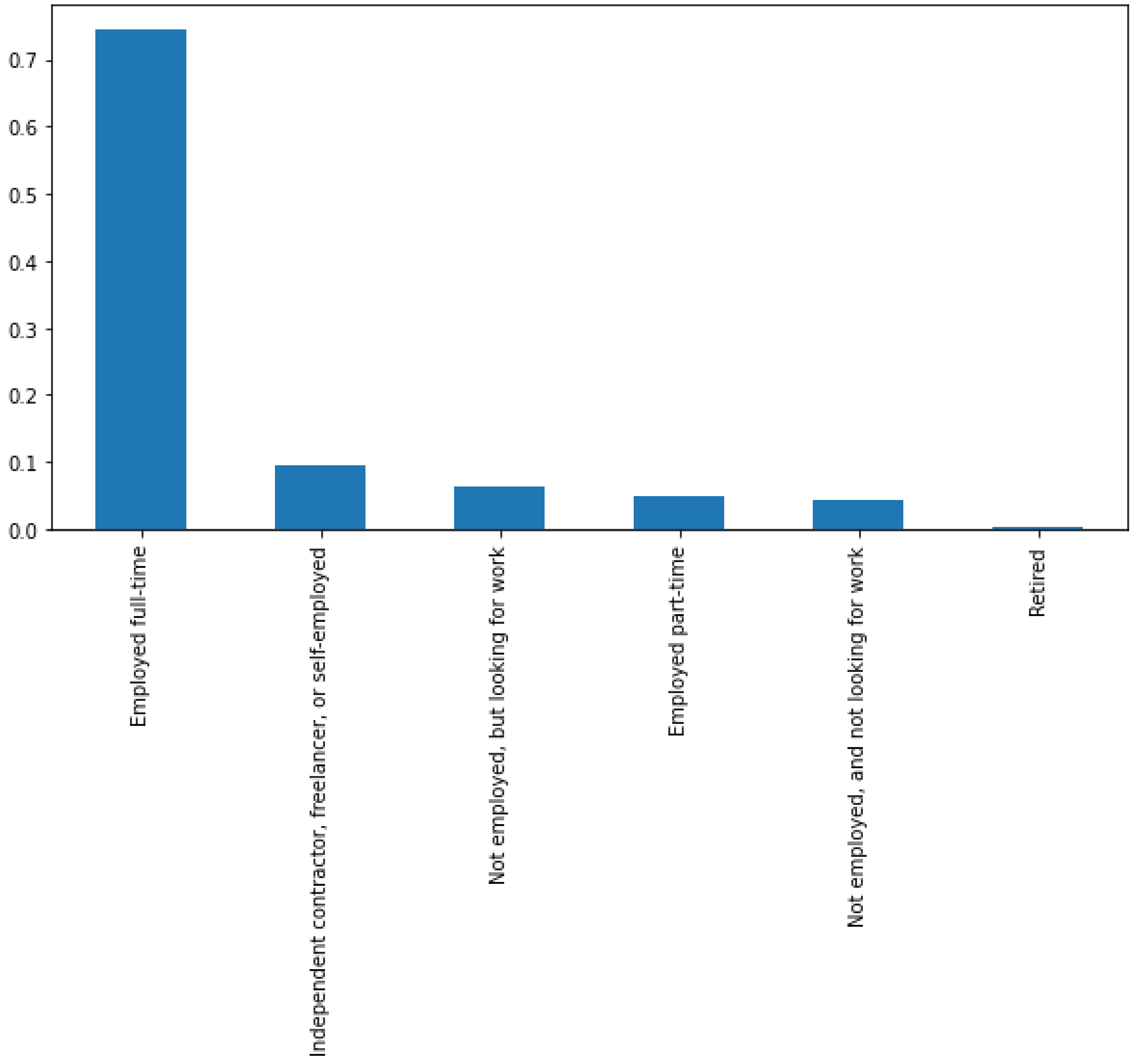


In [162…

```
df2['Employment'].value_counts(normalize=True).plot(kind='bar',figsize=(10,5))
```

Out[162…

```
<AxesSubplot:>
```



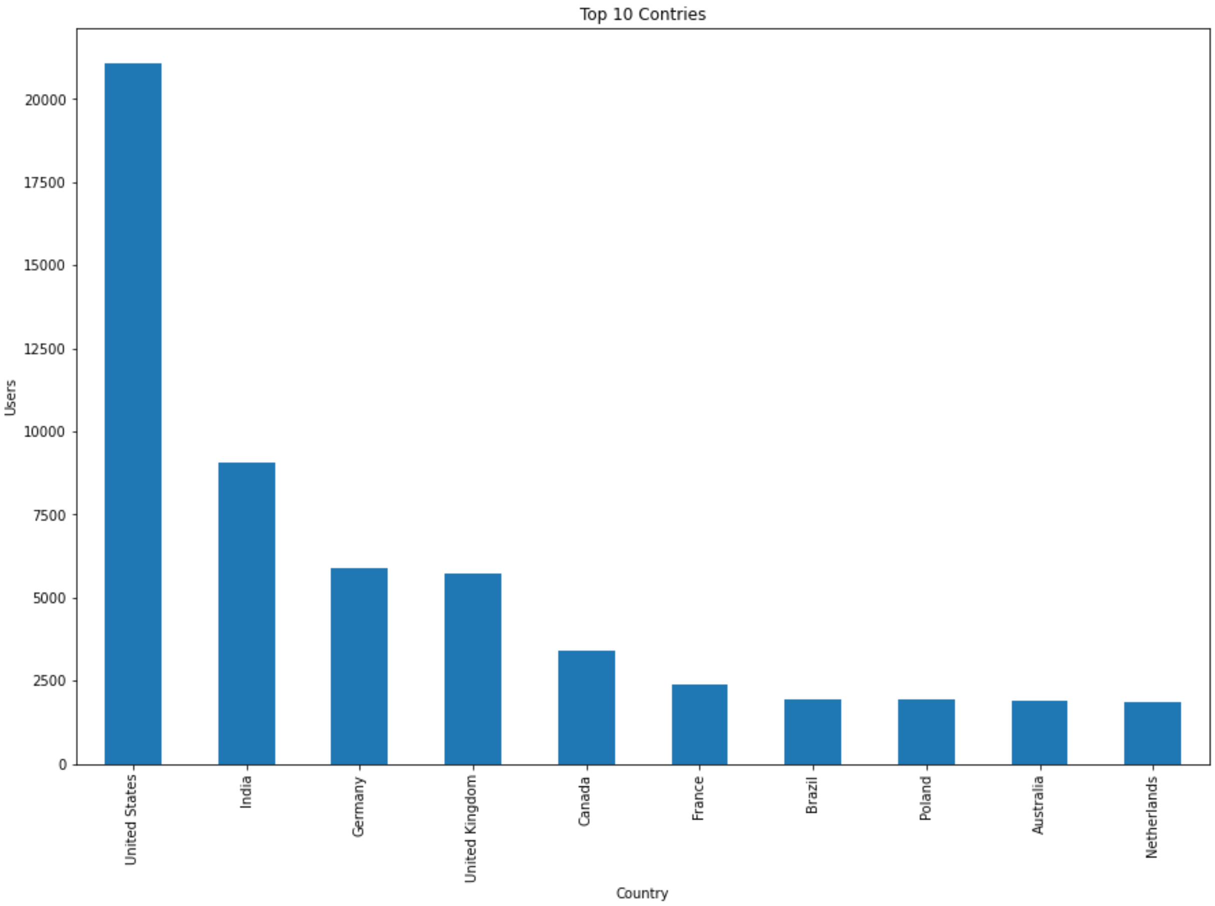
70% Users in this survey are full time employees.

In [185…

```
df2['Country'].value_counts().head(10).plot(kind='bar',figsize=(15,10),xlabel='Count
plt.title('Top 10 Contries')
```

Out[185…

```
Text(0.5, 1.0, 'Top 10 Contries')
```



In [186…

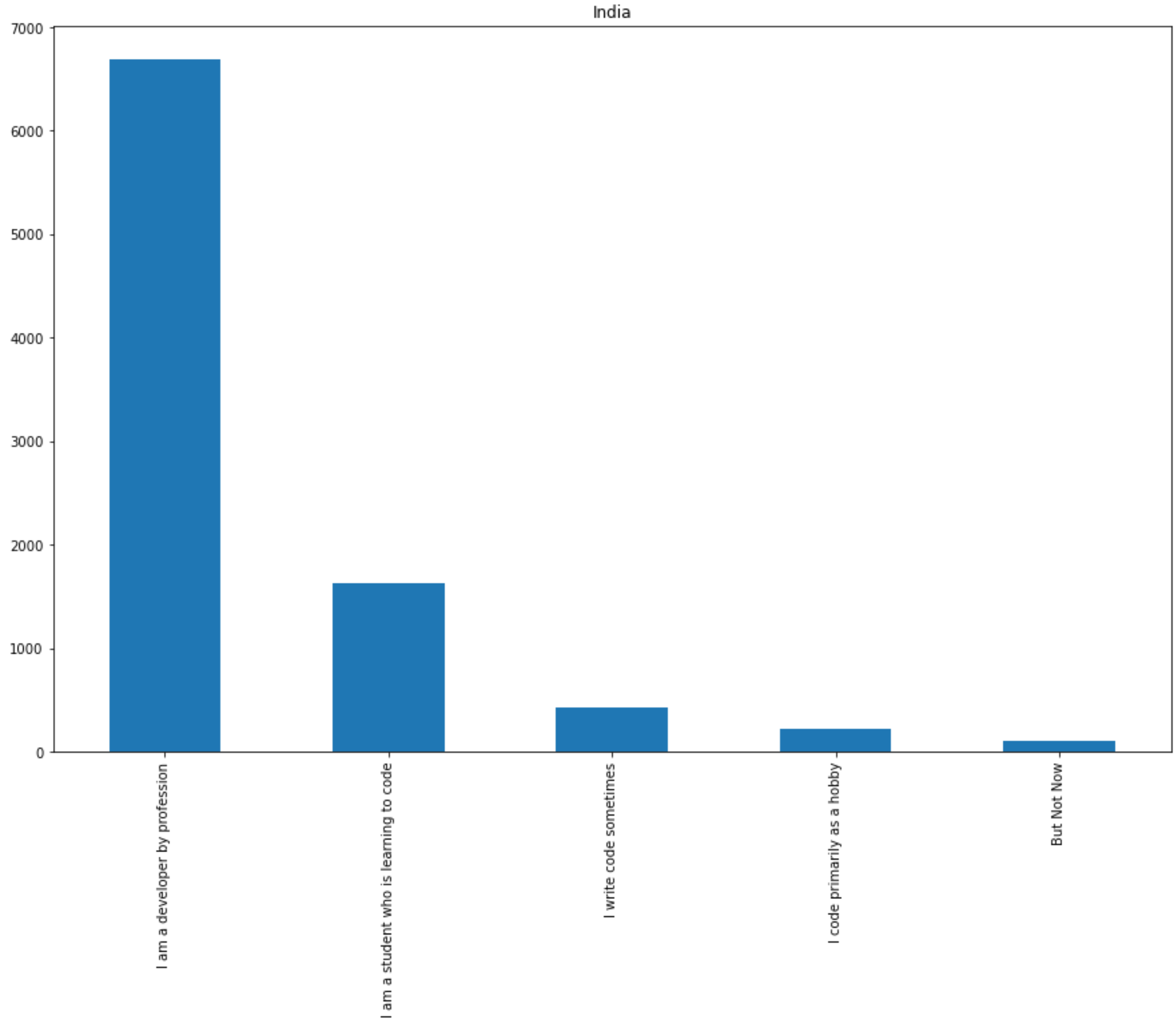
```
country_grp = df2.groupby('Country')
```

In [193…

```
country_grp.get_group('India')['MainBranch'].value_counts().plot(kind='bar',figsize=
plt.title('India')
```

Out[193…

```
Text(0.5, 1.0, 'India')
```

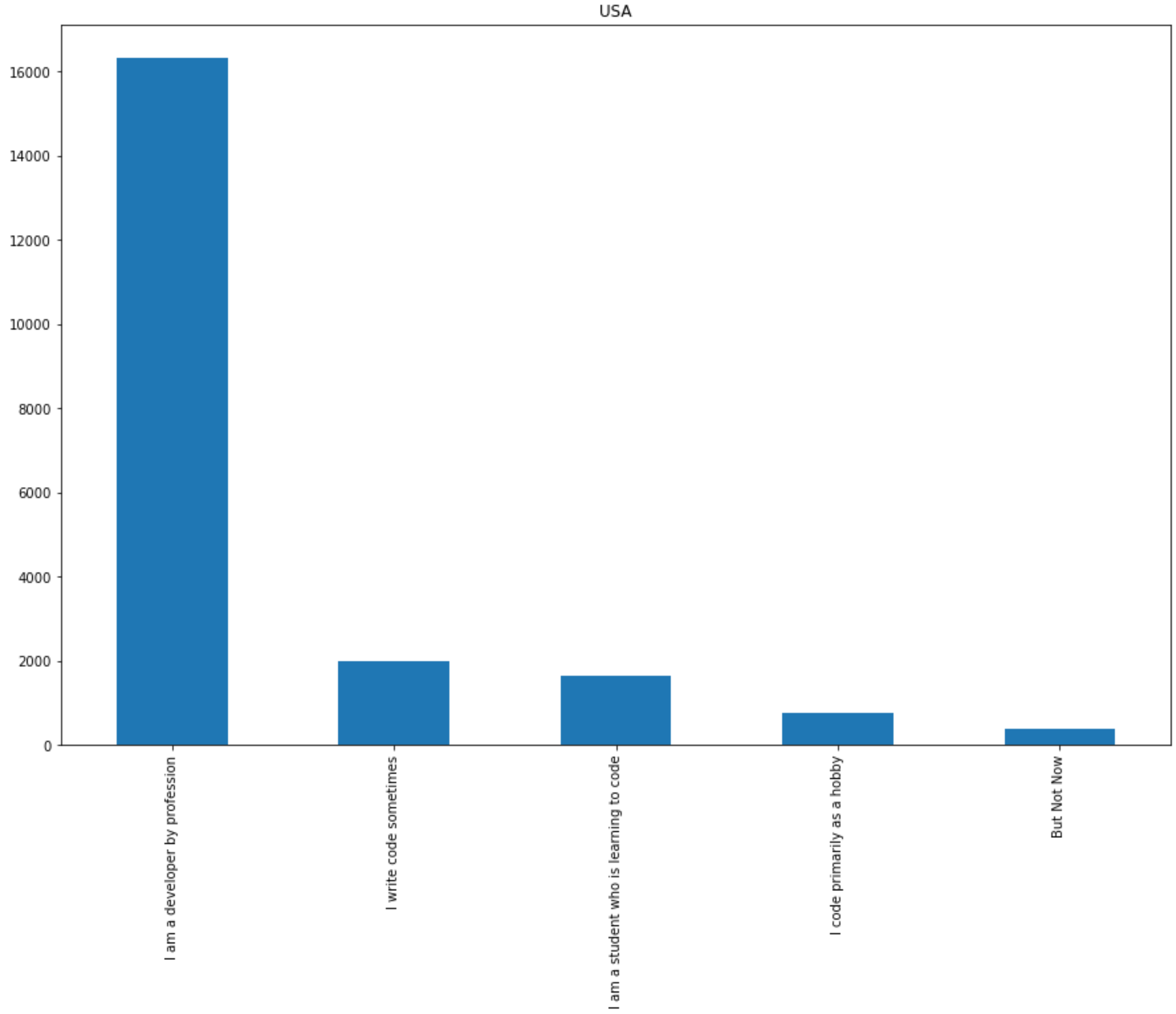


In [219…

```
country_grp.get_group('United States')['MainBranch'].value_counts().plot(kind='bar',
plt.title('USA')
```

Out[219…

```
Text(0.5, 1.0, 'USA')
```

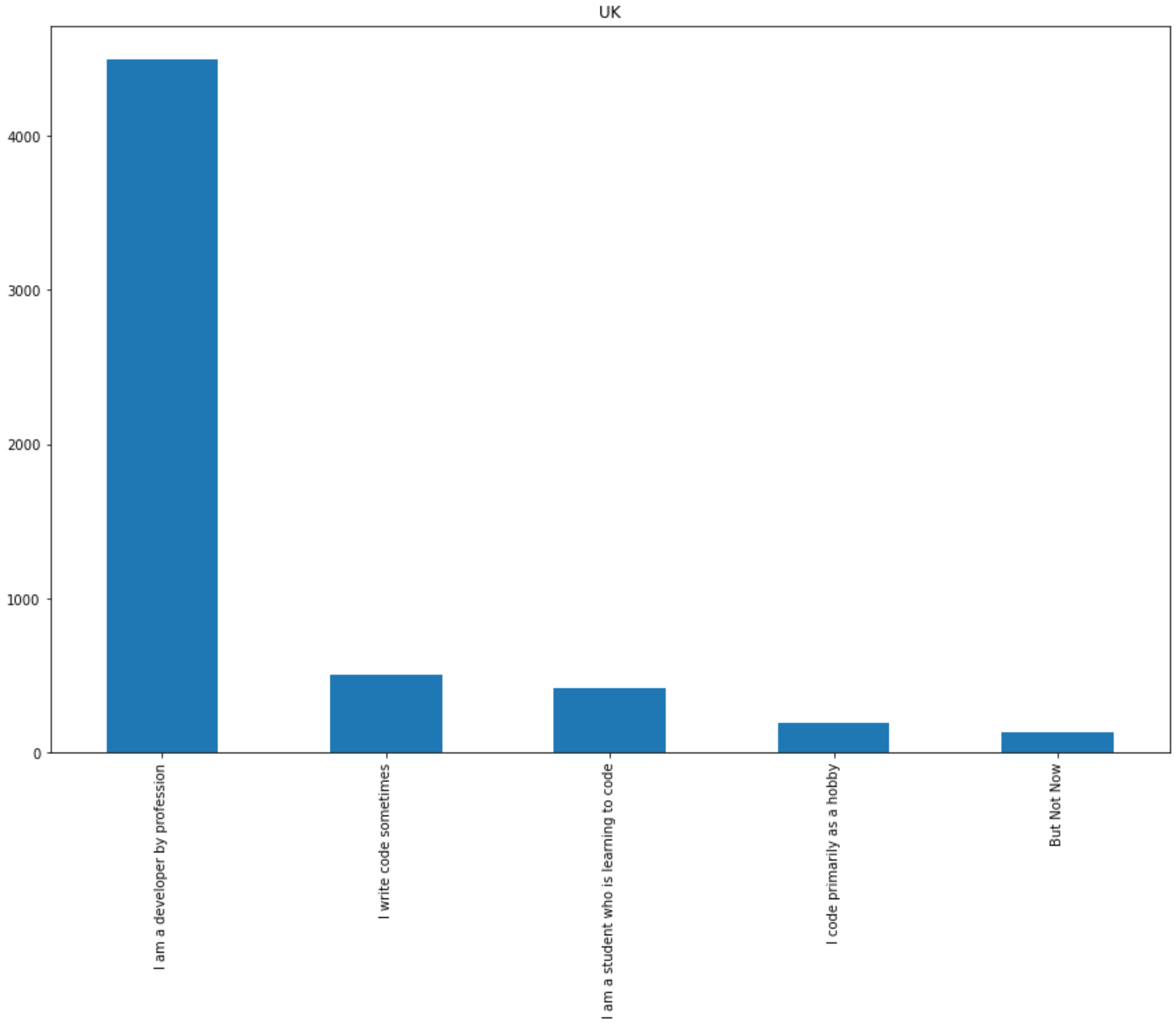


In [220…

```
country_grp.get_group('United Kingdom')['MainBranch'].value_counts().plot(kind='bar'
plt.title('UK')
```

Out[220…

```
Text(0.5, 1.0, 'UK')
```



StackOver flow mostly used by professionals and Students

In [236…

```
df2['EdLevel'].value_counts()
```

Out[236…

```
Bachelor’s degree (BA, BS, B.Eng., etc.)
41627
Master’s degree (MA, MS, M.Eng., MBA, etc.)
19569
Some college/university study without earning a degree
10502
Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)
8642
Associate degree
2938
Other doctoral degree (Ph.D, Ed.D., etc.)
2432
Primary/elementary school
1422
Professional degree (JD, MD, etc.)
1198
I never completed any formal education
553
Name: EdLevel, dtype: int64
```

In [237…

```
df2['EdLevel'].replace('Some college/university study without earning a degree','Dro
df2['EdLevel'].replace('Secondary school (e.g. American high school, German Realschu
df2['EdLevel'].replace('I never completed any formal education','No formal education
df2['EdLevel'].replace('Other doctoral degree (Ph.D, Ed.D., etc.)','doctoral degree'
```

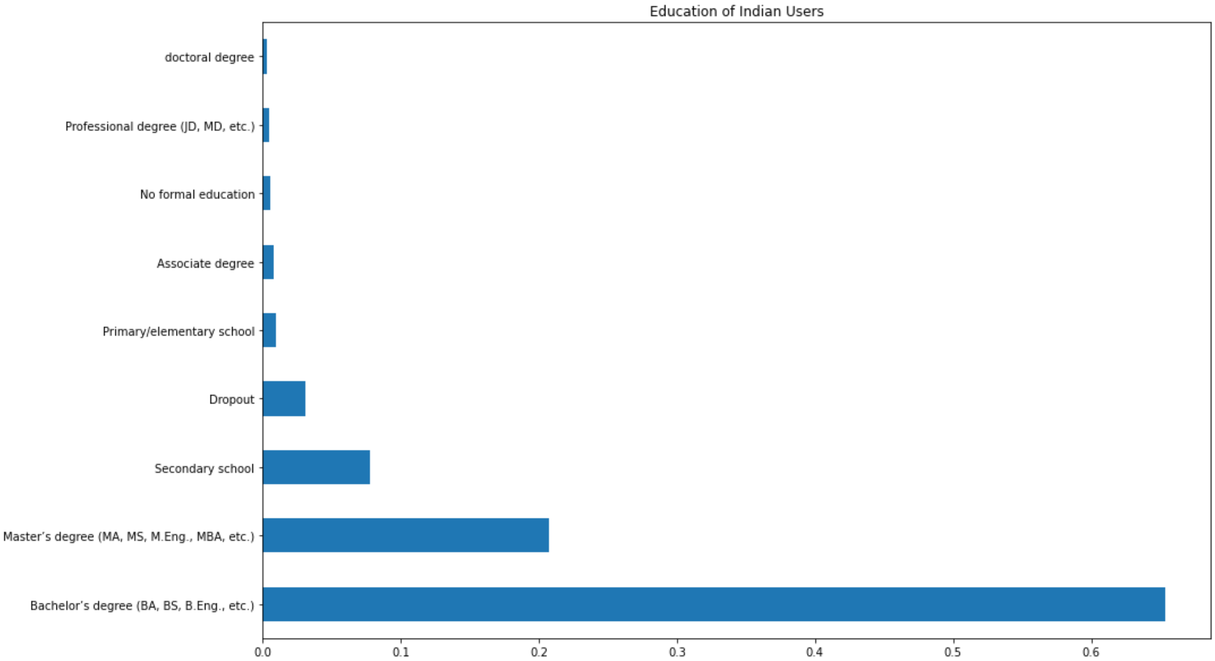
Education Level of Top 5 Countries Users

In [243…

```
country_grp.get_group('India')['EdLevel'].value_counts(normalize=True).plot(kind='ba
plt.title('Education of Indian Users')
```

Out[243…

```
Text(0.5, 1.0, 'Education of Indian Users')
```

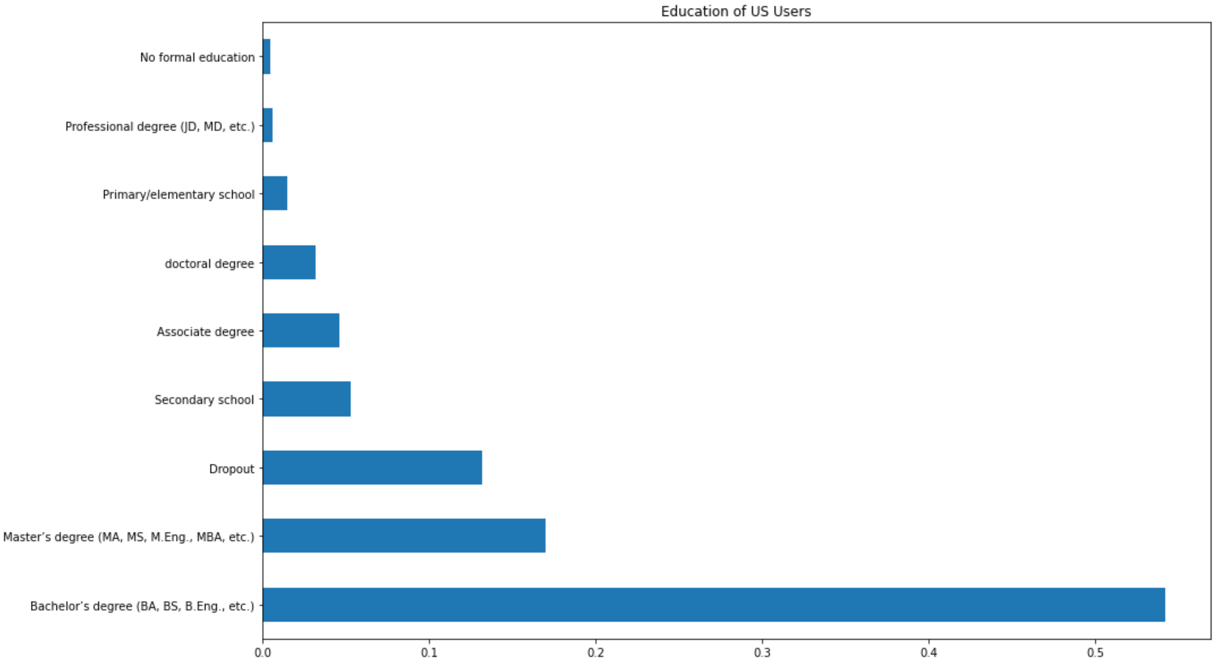


In [244…

```
country_grp.get_group('United States')['EdLevel'].value_counts(normalize=True).plot(
plt.title('Education of US Users')
```

Out[244…

```
Text(0.5, 1.0, 'Education of US Users')
```

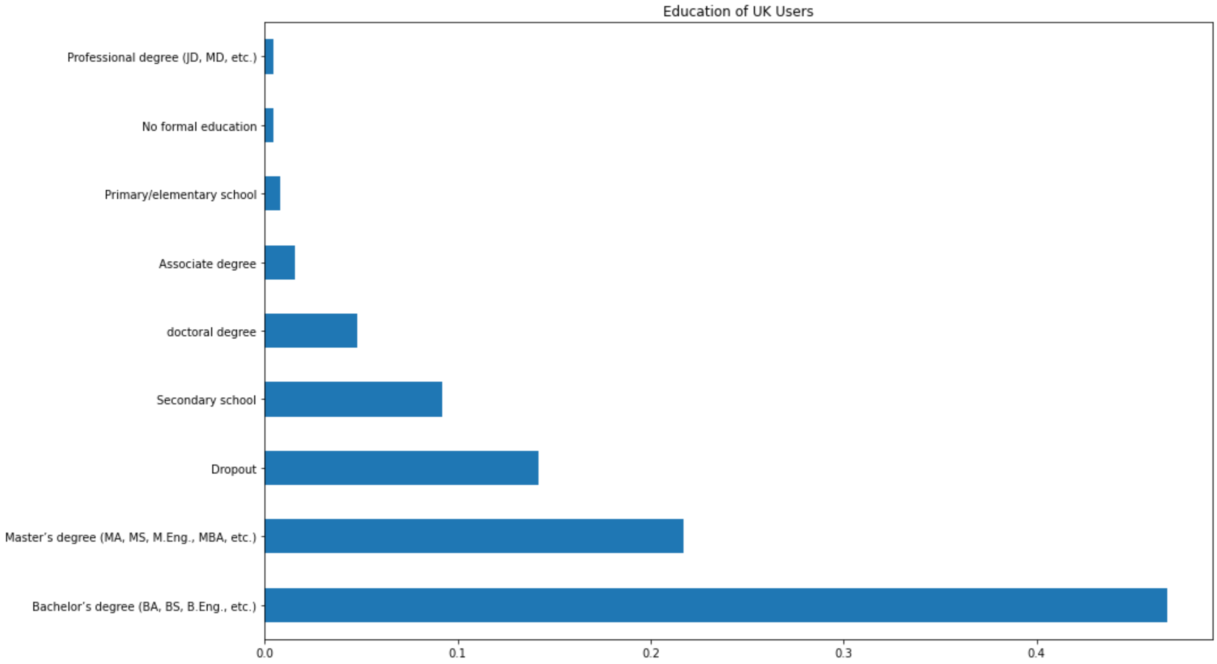


In [245…

```
country_grp.get_group('United Kingdom')['EdLevel'].value_counts(normalize=True).plot
plt.title('Education of UK Users')
```

Out[245…

```
Text(0.5, 1.0, 'Education of UK Users')
```

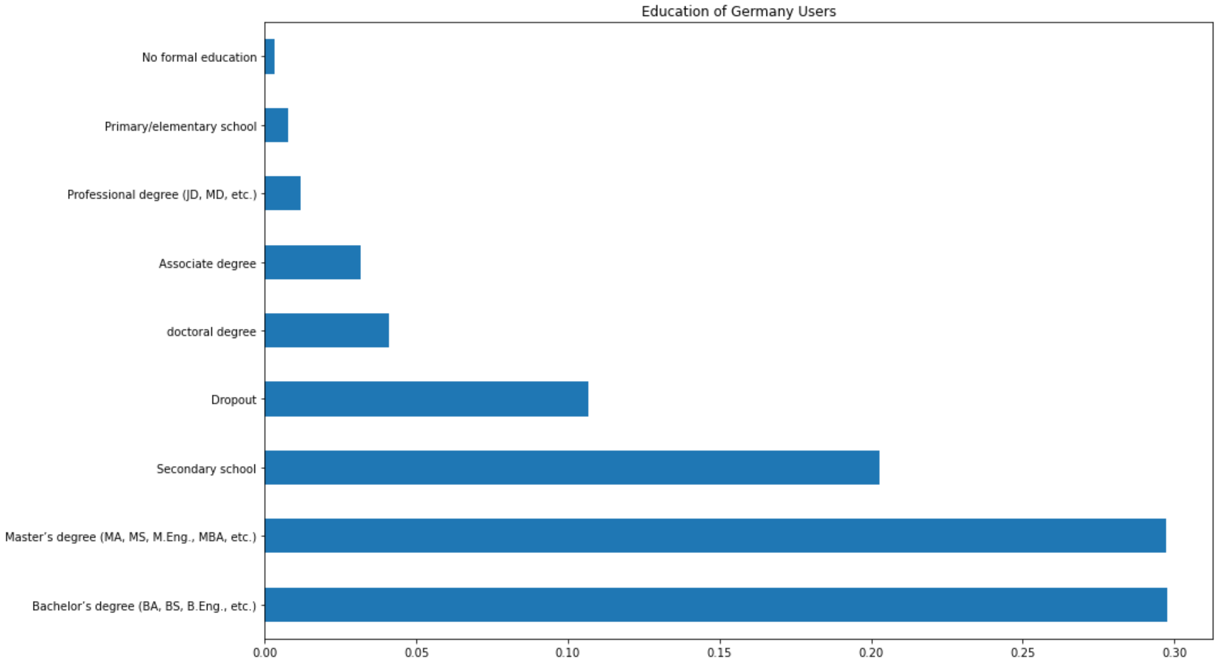


In [246…

```
country_grp.get_group('Germany')['EdLevel'].value_counts(normalize=True).plot(kind='
plt.title('Education of Germany Users')
```

Out[246…

```
Text(0.5, 1.0, 'Education of Germany Users')
```

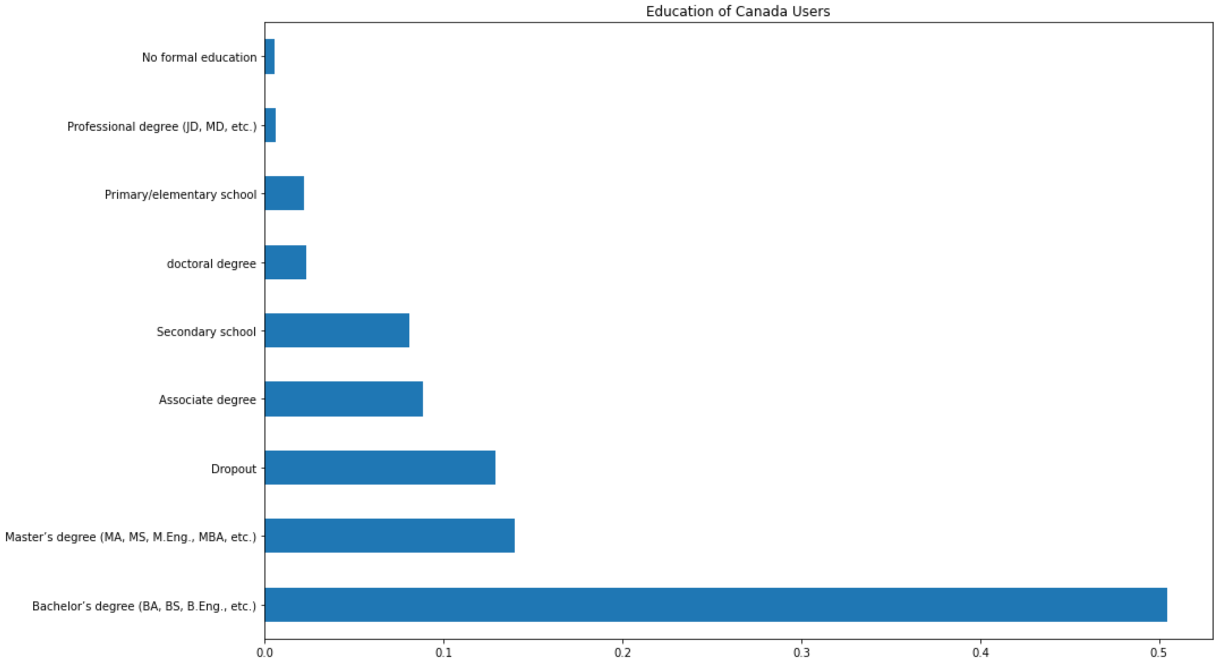


In [247…

```
country_grp.get_group('Canada')['EdLevel'].value_counts(normalize=True).plot(kind='b
plt.title('Education of Canada Users')
```

Out[247…

```
Text(0.5, 1.0, 'Education of Canada Users')
```



Most Country's users have Bachelor's Degree Followed by Masters. In Germnay Bachelor's holder and masters % is almost same.Noticed How dropouts are more in USA,Canada and UK

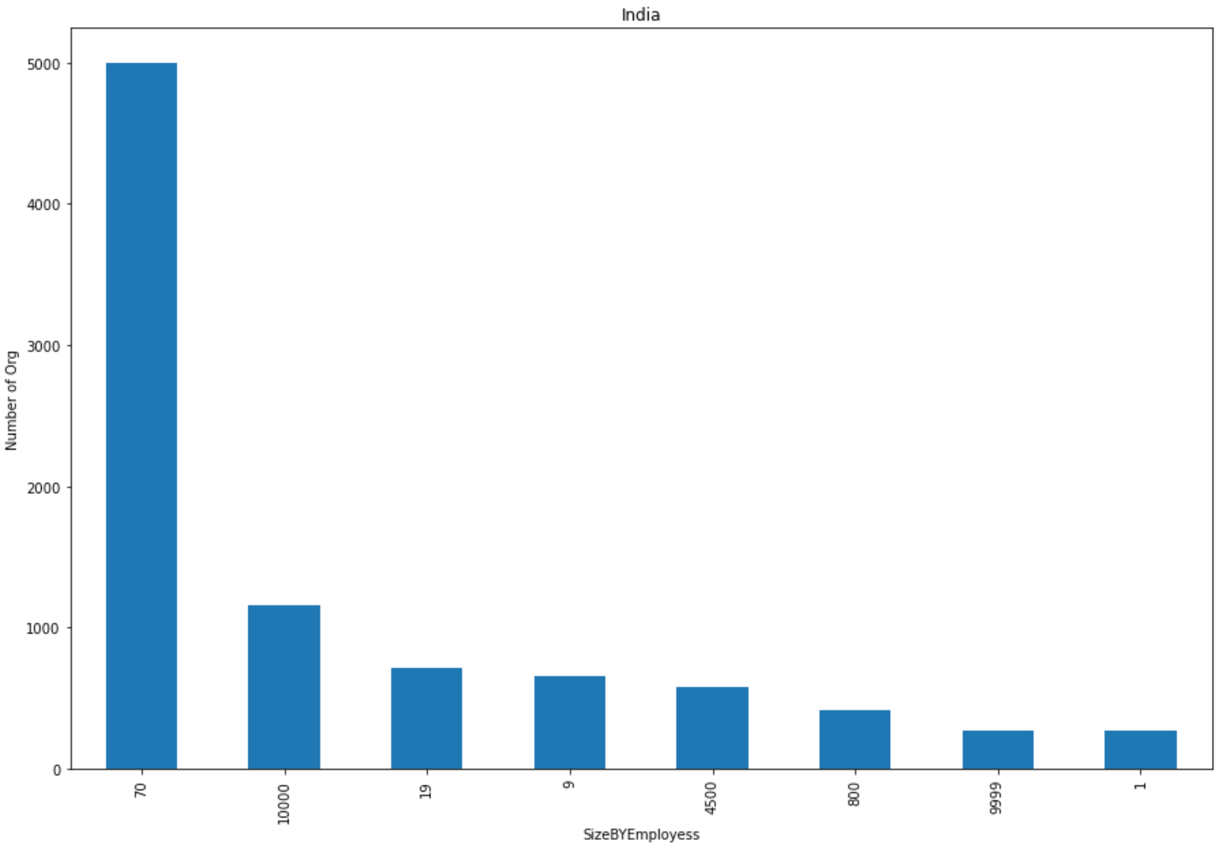
Organisation size by top countres.

In [264…

```
country_grp.get_group('India')['OrgSize'].value_counts().plot(kind='bar',figsize=(15
                                                              xlabel='SizeBYEmployes
plt.title('India')
```

Out[264…

```
Text(0.5, 1.0, 'India')
```

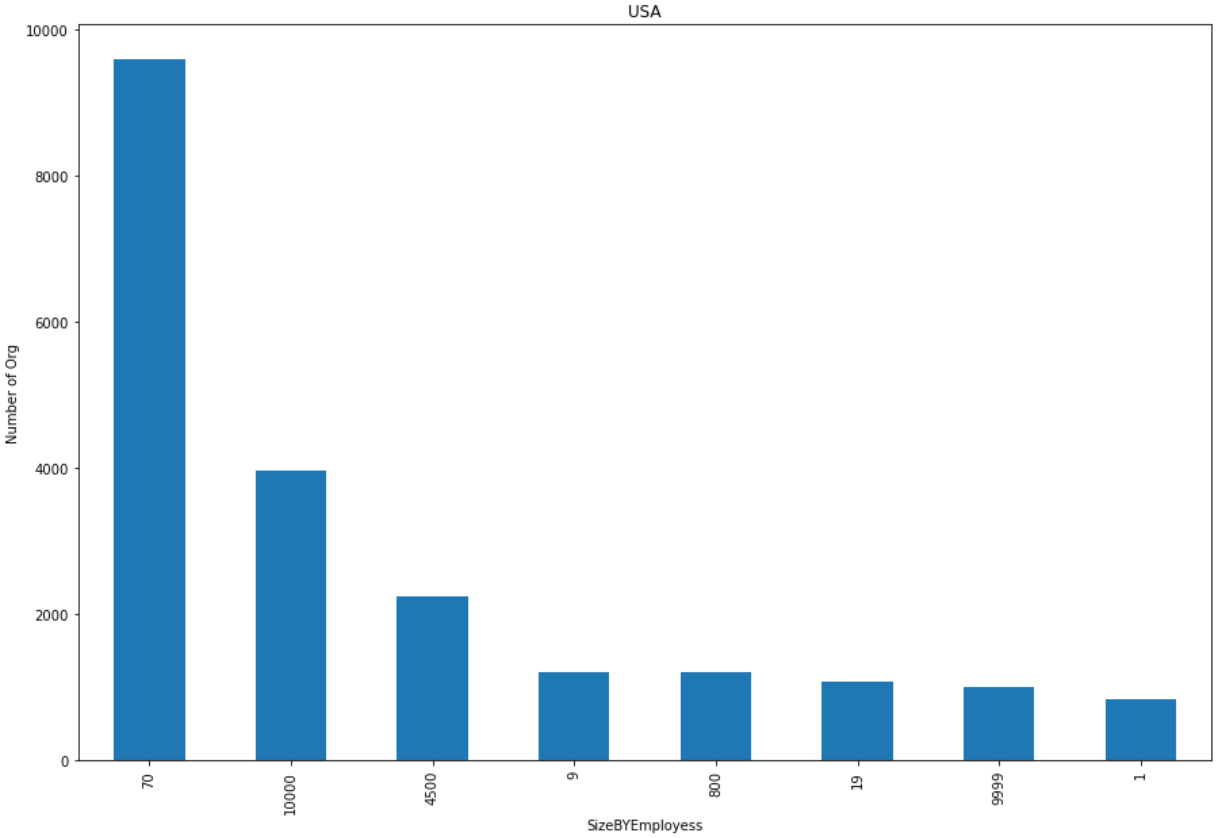


In [265…

```
country_grp.get_group('United States')['OrgSize'].value_counts().plot(kind='bar',fig
                                                               xlabel='SizeBYEmployes
plt.title('USA')
```

Out[265…

```
Text(0.5, 1.0, 'USA')
```

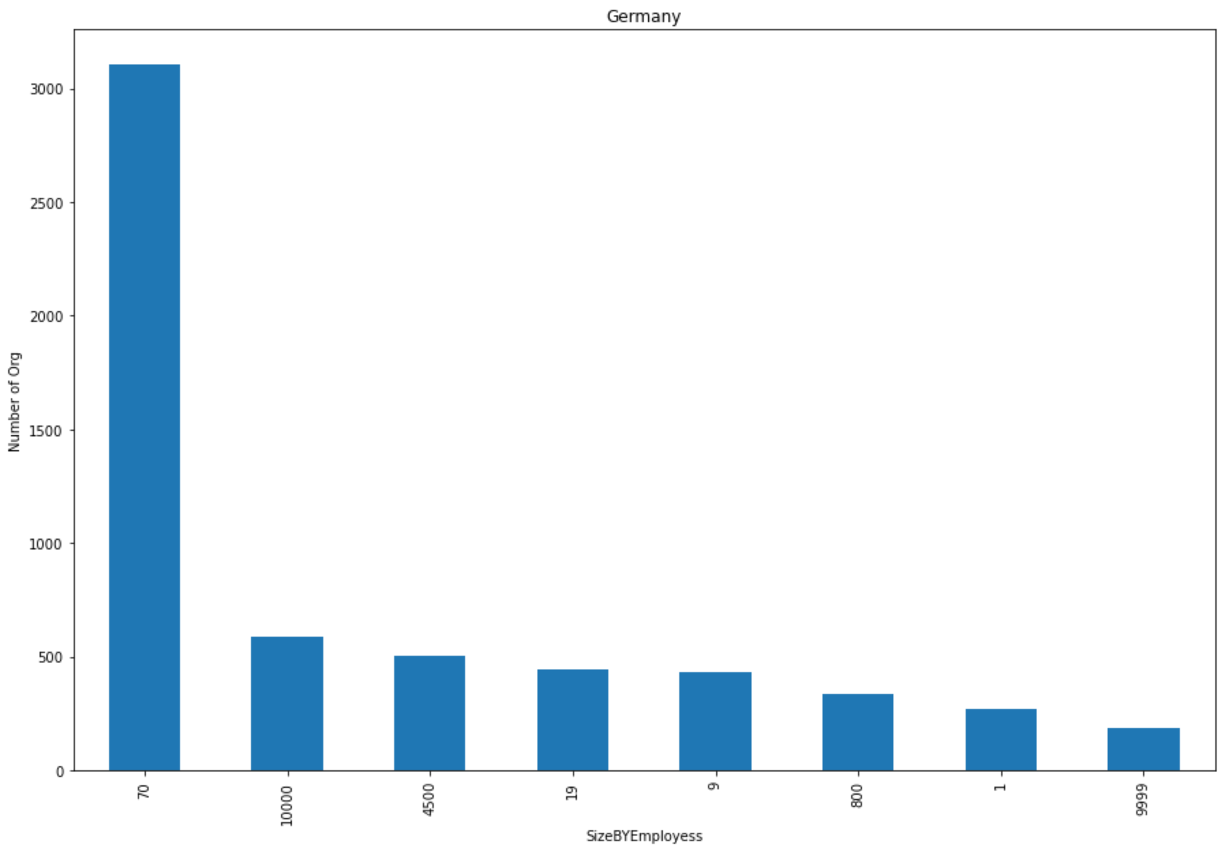


In [266…

```
country_grp.get_group('Germany')['OrgSize'].value_counts().plot(kind='bar',figsize=(
                                                               xlabel='SizeBYEmployes
plt.title('Germany')
```

Out[266…

```
Text(0.5, 1.0, 'Germany')
```



This Data not giving Much information because there was many missing values So showing same for every Country

Lets check Age Data by country

In [278…

```
country_grp.get_group('United States')['YearsCode'].describe() #agg(['mean','median'
```

Out[278…

```
count    21081.000000
mean        13.905365
std         10.435717
min          0.000000
25%          6.000000
50%         10.000000
75%         20.000000
max         51.000000
Name: YearsCode, dtype: float64
```

In [288…

```
print('USA\n--------------------------------------------')
print(country_grp.get_group('United States')['YearsCode'].describe())
print('India\n------------------------------------------')
print(country_grp.get_group('India')['YearsCode'].describe())
print('Germany\n----------------------------------------')
print(country_grp.get_group('Germany')['YearsCode'].describe())
print('Canada\n---------------------------------------------')
print(country_grp.get_group('Canada')['YearsCode'].describe())
print('United Kingdom\n-------------------------------------')
print(country_grp.get_group('United Kingdom')['YearsCode'].describe())
```

```
USA
--------------------------------------------
count    21081.000000
mean        13.905365
std         10.435717
min          0.000000
```

```
25%           6.000000
50%          10.000000
75%          20.000000
max          51.000000
Name: YearsCode, dtype: float64
India
------------------------------------------
count    9061.000000
mean        6.605452
std         4.589946
min         0.000000
25%         3.000000
50%         6.000000
75%         9.000000
max        51.000000
Name: YearsCode, dtype: float64
Germany
----------------------------------------
count    5866.000000
mean       12.758950
std         8.598292
min         0.000000
25%         6.000000
50%        10.000000
75%        17.000000
max        51.000000
Name: YearsCode, dtype: float64
Canada
--------------------------------------------
count    3395.000000
mean       13.232695
std         9.830924
min         0.000000
25%         6.000000
50%        10.000000
75%        19.000000
max        51.000000
Name: YearsCode, dtype: float64
United Kingdom
-------------------------------------
count    5737.000000
mean       14.708384
std        10.678998
min         0.000000
25%         6.000000
50%        12.000000
75%        20.000000
max        51.000000
Name: YearsCode, dtype: float64
```

In [301…

```
print(country_grp.get_group('India')['YearsCodePro'].agg(['mean','max','min']))
print(country_grp.get_group('United States')['YearsCodePro'].agg(['mean','max','min'
```

```
mean      5.004083
max      51.000000
min       0.000000
Name: YearsCodePro, dtype: float64
mean      9.410986
max      51.000000
min       0.000000
Name: YearsCodePro, dtype: float64
```

Not geeting much information from these columns

In [303…

```
df2['JobSeek'].replace('I’m not actively looking, but I am open to new opportunities
df2['JobSeek'].replace('I am not interested in new job opportunities','I am not inte
```

In [ ]:

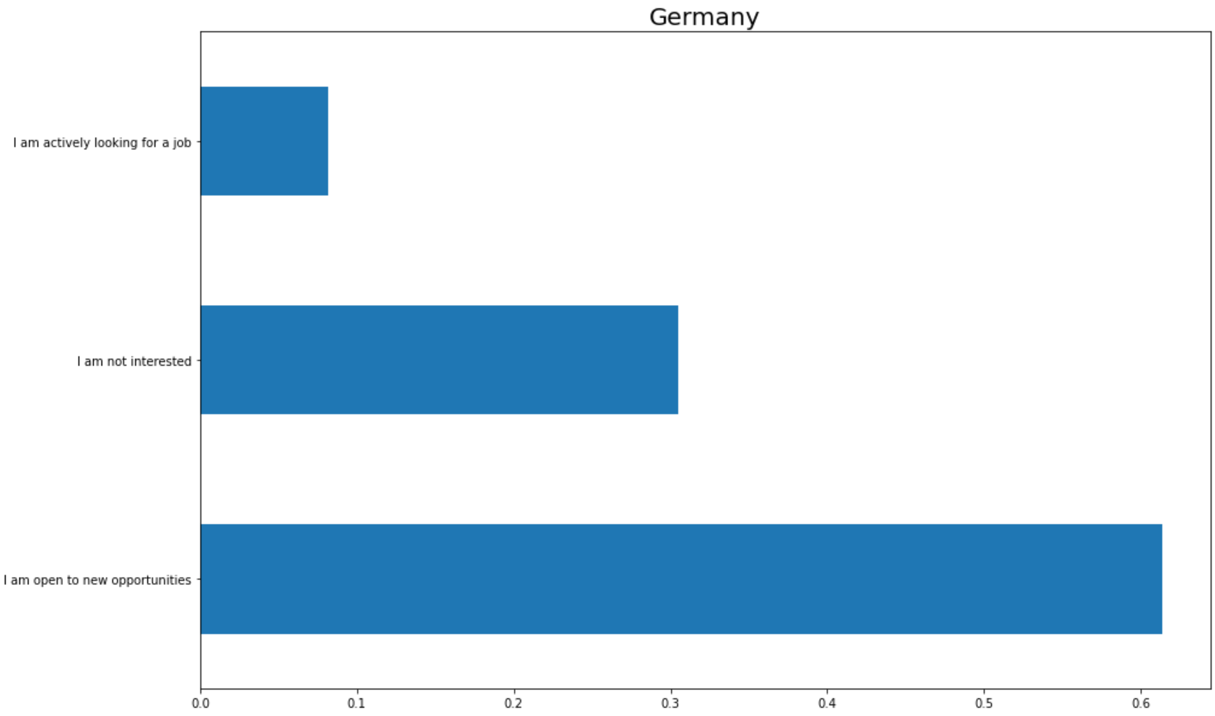
```
### Check if users looking for job or not by country
```

In [309…

```
country_grp.get_group('Germany')['JobSeek'].value_counts(normalize=True).plot(kind='
plt.title('Germany',fontsize=20)
```

Out[309…

```
Text(0.5, 1.0, 'Germany')
```

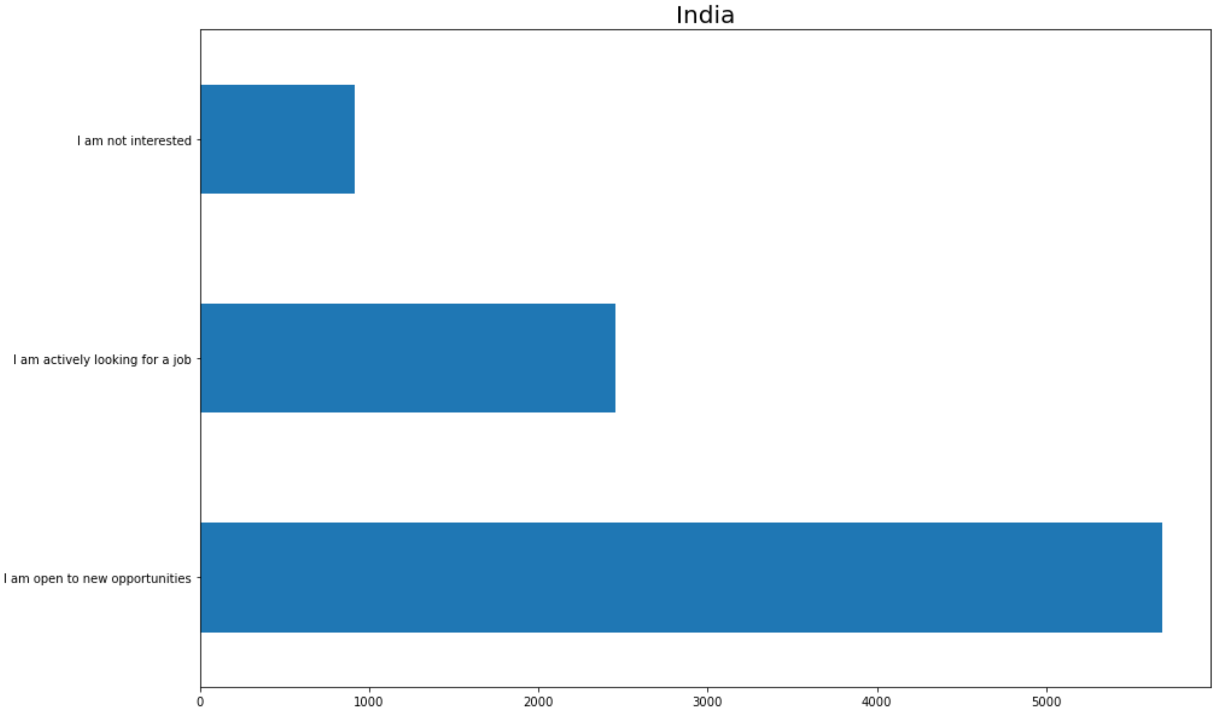


In [311…

```
country_grp.get_group('India')['JobSeek'].value_counts().plot(kind='barh',figsize=(1
plt.title('India',fontsize=20)
```

Out[311…

```
Text(0.5, 1.0, 'India')
```

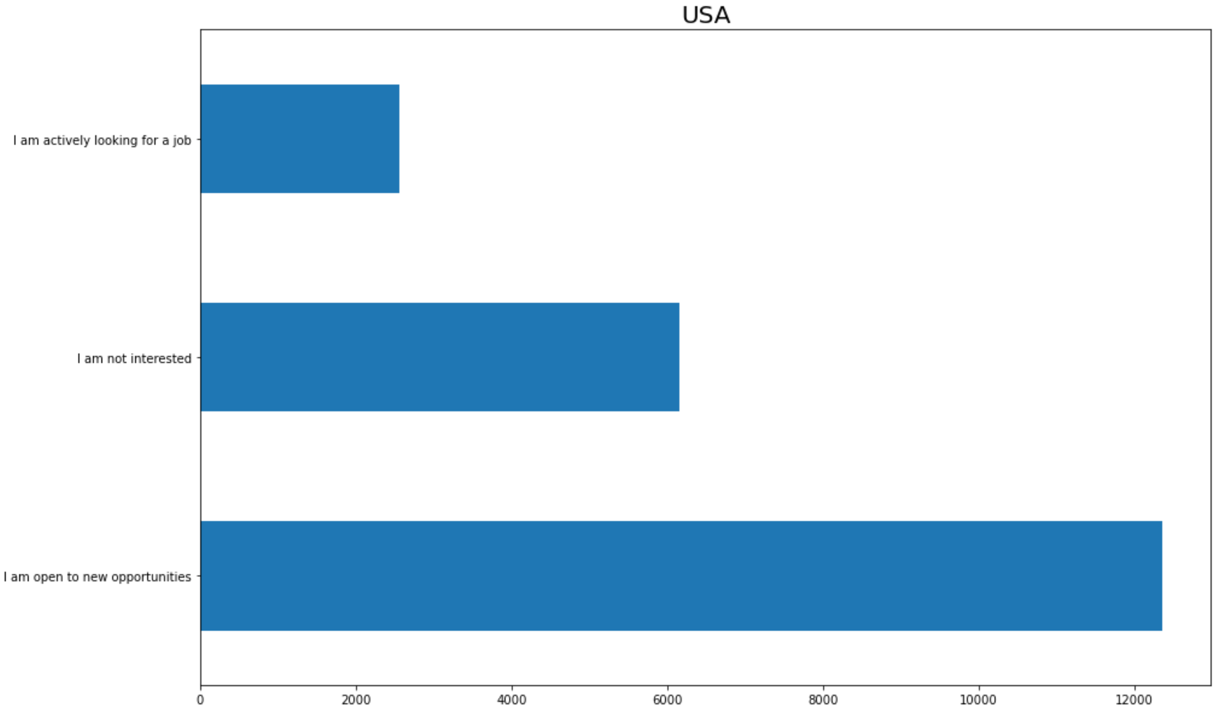


In [312…

```
country_grp.get_group('United States')['JobSeek'].value_counts().plot(kind='barh',fi
plt.title('USA',fontsize=20)
```

Out[312…

```
Text(0.5, 1.0, 'USA')
```

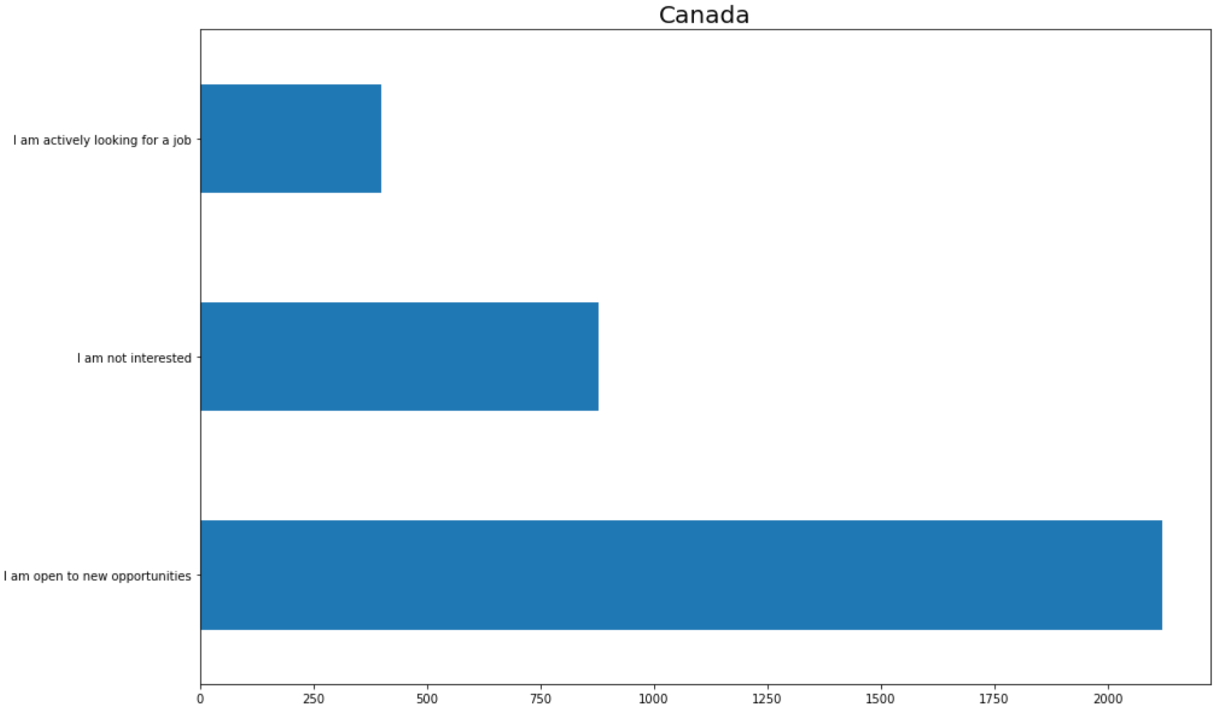


In [313…

```
country_grp.get_group('Canada')['JobSeek'].value_counts().plot(kind='barh',figsize=(
plt.title('Canada',fontsize=20)
```

Out[313…

```
Text(0.5, 1.0, 'Canada')
```

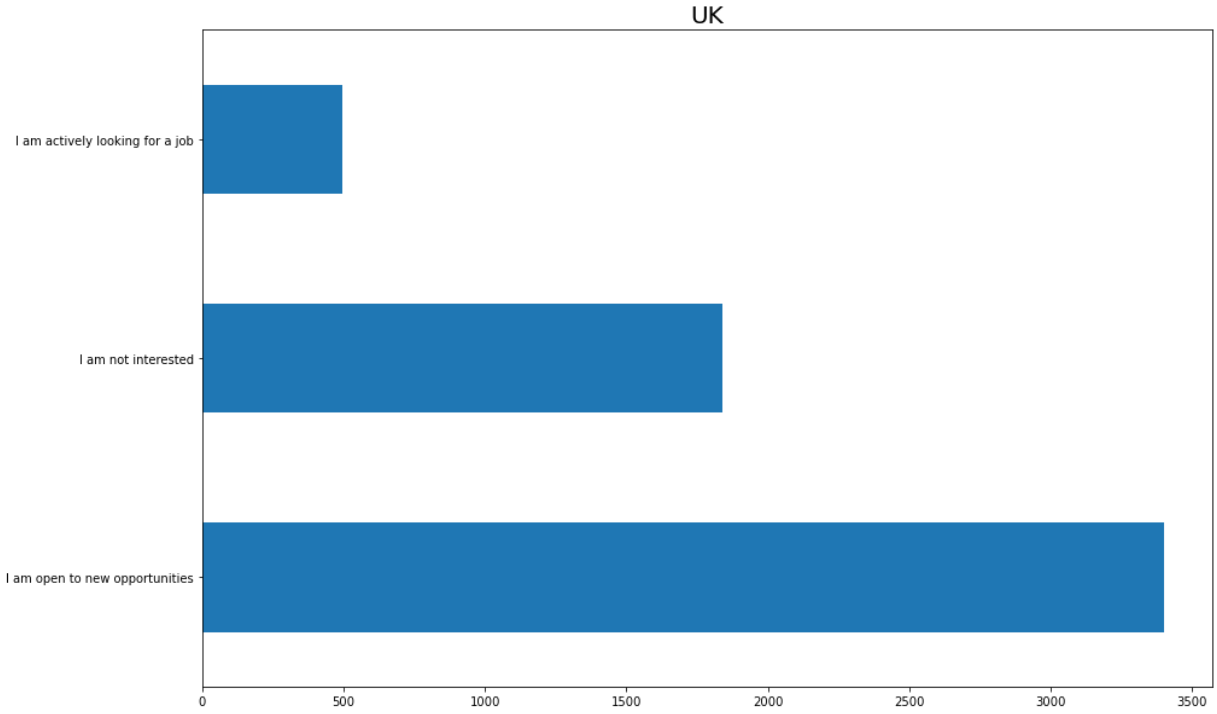


In [314…

```
country_grp.get_group('United Kingdom')['JobSeek'].value_counts().plot(kind='barh',f
plt.title('UK',fontsize=20)
```

Out[314…

```
Text(0.5, 1.0, 'UK')
```



In most of the countries users already working, followed by not interested but in India people also looking for work.

Lets check the salary accross the countires

In [331…

```
print('India\n------------------------------')
round(country_grp.get_group('India')['ConvertedComp'].describe(percentiles=[0.01,0.0
```

Out[331…

```
India
------------------------------
count       9061.0
mean       44387.0
std        58706.0
min            0.0
1%           316.0
3%          2016.0
5%          2940.0
50%        57287.0
95%        57287.0
96%        57287.0
97%        67164.0
98%       115883.0
99%       218292.0
max      2000000.0
Name: ConvertedComp, dtype: float64
```

In [332…

```
print('USA\n-----------------------------------')
round(country_grp.get_group('United States')['ConvertedComp'].describe(percentiles=[
```

Out[332…

```
USA
-----------------------------------
count      21081.0
mean      193914.0
std       390962.0
min            0.0
1%         15000.0
3%         40000.0
5%         50000.0
50%        85000.0
```

```
95%      1080000.0
96%      1440000.0
97%      2000000.0
98%      2000000.0
99%      2000000.0
max      2000000.0
Name: ConvertedComp, dtype: float64
```

In [330…

```
print('Germany\n-----------------------------------')
round(country_grp.get_group('Germany')['ConvertedComp'].describe(percentiles=[0.01,0
```

```
Germany
-----------------------------------
```

Out[330…

```
count       5866.0
mean       90758.0
std       152419.0
min            0.0
1%          7053.0
3%         12372.0
5%         16968.0
50%        57287.0
95%       170100.0
96%       412464.0
97%       687444.0
98%       783696.0
99%       962424.0
100%     2000000.0
max      2000000.0
Name: ConvertedComp, dtype: float64
```

In [333…

```
print('Canada\n-----------------------------------')
round(country_grp.get_group('Canada')['ConvertedComp'].describe(percentiles=[0.01,0.
```

```
Canada
-----------------------------------
```

Out[333…

```
count       3395.0
mean      108298.0
std       190389.0
min            0.0
1%          8899.0
3%         27500.0
5%         34352.0
50%        57287.0
95%       371002.0
96%       727049.0
97%      1000000.0
98%      1000000.0
99%      1000000.0
max      2000000.0
Name: ConvertedComp, dtype: float64
```

In [334…

```
print('UK\n-----------------------------------')
round(country_grp.get_group('United Kingdom')['ConvertedComp'].describe(percentiles=
```

```
UK
-----------------------------------
```

Out[334…

```
count       5737.0
mean      133857.0
std       210150.0
min            0.0
1%         16005.0
3%         26169.0
```

```
5%          31403.0
50%         57287.0
95%        706572.0
96%        785088.0
97%        863592.0
98%        973500.0
99%       1000000.0
max       2000000.0
Name: ConvertedComp, dtype: float64
```

There is outliers but if, we ignore it salaries are more in USA,UK,Germany and Canada. India is far behind

In [347…

```
country_grp.get_group('India')['WorkWeekHrs'].agg(['mean','median','max','min'])
```

Out[347…

```
mean        41.354575
median      40.000000
max       4850.000000
min          2.000000
Name: WorkWeekHrs, dtype: float64
```

In [349…

```
country_grp.get_group('United States')['WorkWeekHrs'].agg(['mean','median','max','mi
```

Out[349…

```
mean       41.641599
median     40.000000
max       168.000000
min         1.000000
Name: WorkWeekHrs, dtype: float64
```

In [365…

```
country_grp.get_group('Germany')['WorkWeekHrs'].agg(['mean','median','max','min'])
```

Out[365…

```
mean       40.177711
median     40.000000
max       425.000000
min         3.000000
Name: WorkWeekHrs, dtype: float64
```

In [384…

```
country_grp.get_group('United Kingdom')['WorkWeekHrs'].agg(['mean','median','max','m
```

Out[384…

```
mean       39.64756
median     40.00000
max       375.00000
min         3.50000
Name: WorkWeekHrs, dtype: float64
```

Average working hours also around 40 and same in every country

In [385…

```
df2['WorkLoc'].value_counts()
```

Out[385…

```
Office                                            59420
Home                                              23278
Other place, such as a coworking space or cafe     6185
Name: WorkLoc, dtype: int64
```

In [386…

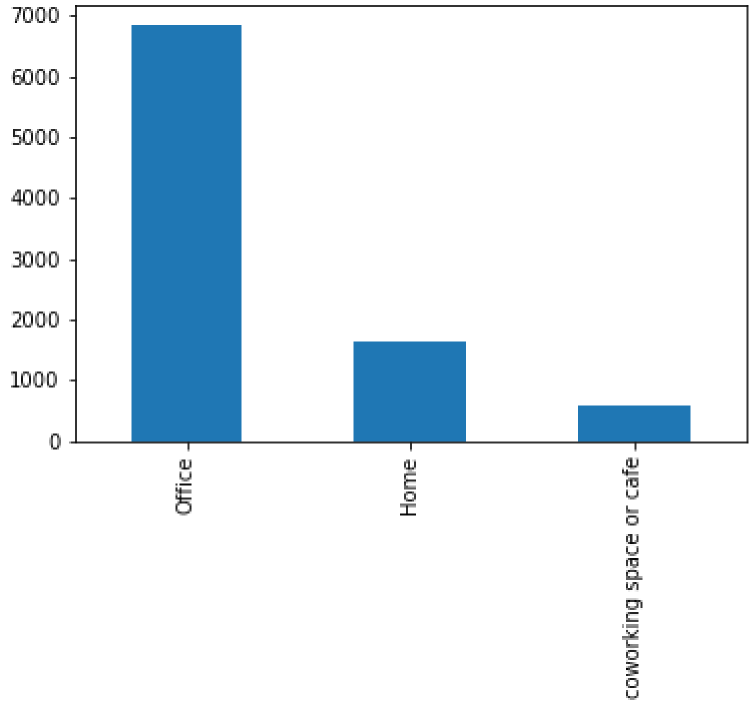
```
df2['WorkLoc'].replace('Other place, such as a coworking space or cafe','coworking s
```

In [388…

```
country_grp.get_group('India')['WorkLoc'].value_counts().plot(kind='bar')
```

```
<AxesSubplot:>
```

Out[388…

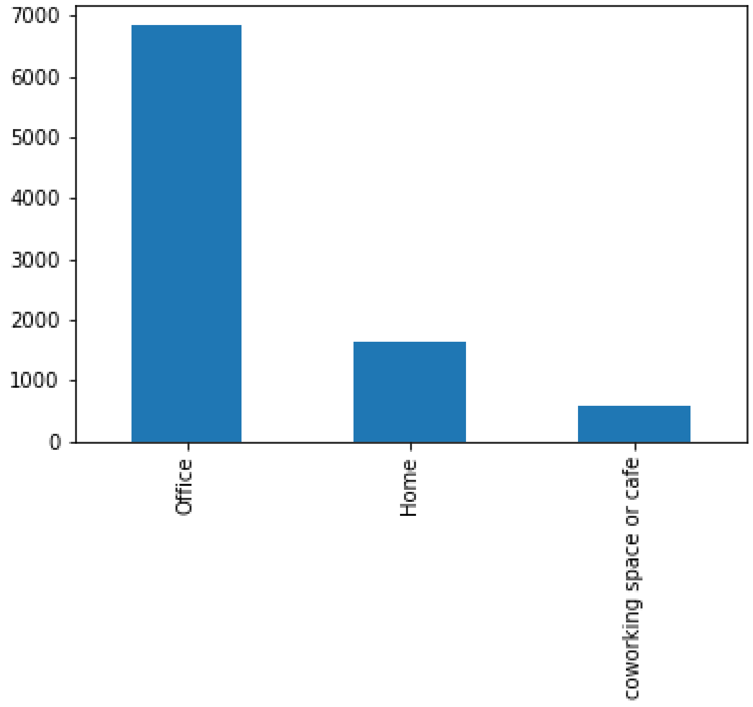


In [389…

```
country_grp.get_group('India')['WorkLoc'].value_counts().plot(kind='bar')
```

Out[389…

```
<AxesSubplot:>
```



As this survey is from 2019 so large number of people used to work from office

Lets check most used programming languages

In [395…

```
country_knows_Python = country_grp['LanguageWorkedWith'].apply(lambda x : x.str.cont
```

In [396…

```
country_knows_Java = country_grp['LanguageWorkedWith'].apply(lambda x : x.str.contai
```

In [397…

```
country_knows_C = country_grp['LanguageWorkedWith'].apply(lambda x : x.str.contains(
```

In [398…

```
respondents = df2['Country'].value_counts()
```

In [416…

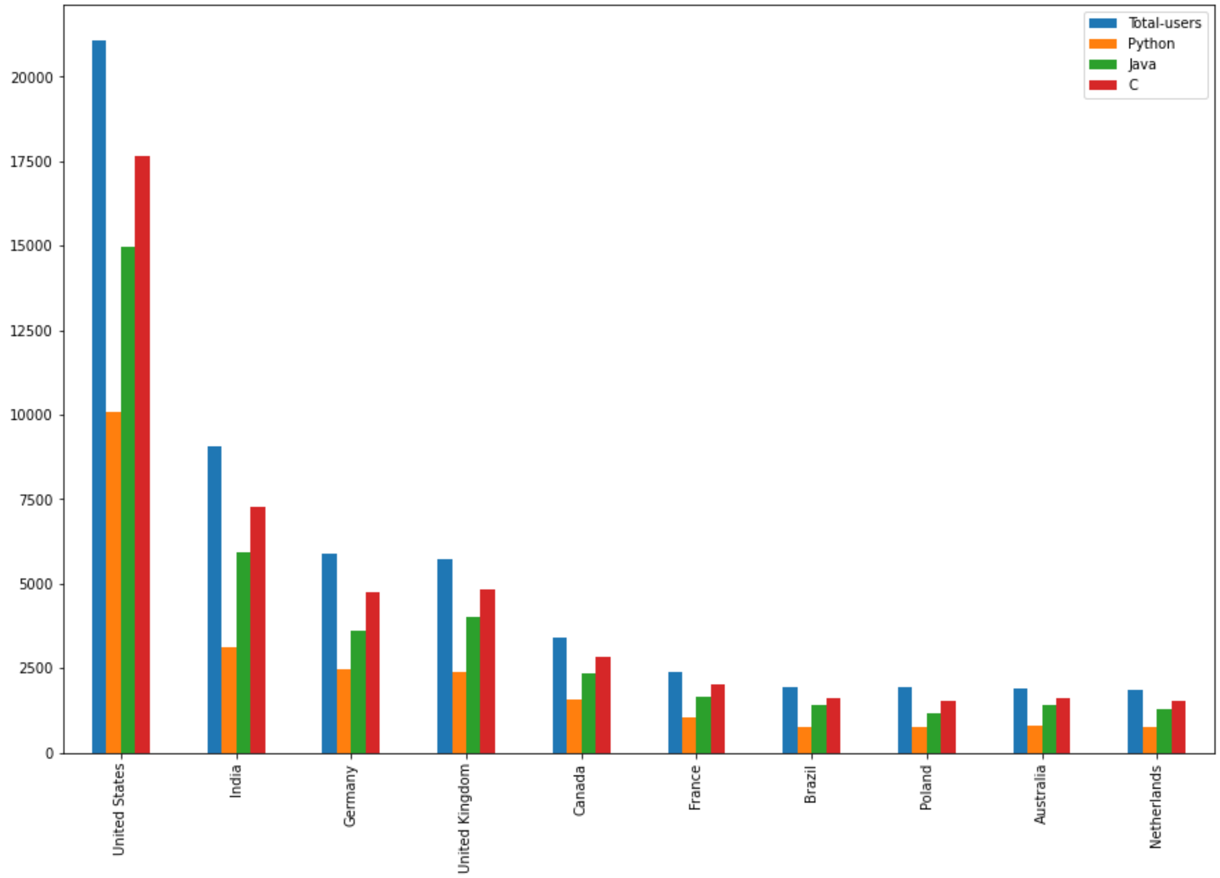
```
Top_3_lang = pd.concat([respondents,country_knows_Python,country_knows_Java,country_
```

In [418…

```
Top_3_lang.head(10).plot(kind='bar',figsize=(15,10))
```

Out[418…

```
<AxesSubplot:>
```



It seems that C++ was most popular in 2019, followed by Javascript. Python was growing at that time.

In [421…

```
pyCharm = country_grp['DevEnviron'].apply(lambda x : x.str.contains('PyCharm').sum()
atom = country_grp['DevEnviron'].apply(lambda x : x.str.contains('Atom').sum())
vs_code =country_grp['DevEnviron'].apply(lambda x : x.str.contains('Visual Studio Co
jupyter = country_grp['DevEnviron'].apply(lambda x : x.str.contains('Jupyter').sum()
```

In [422…

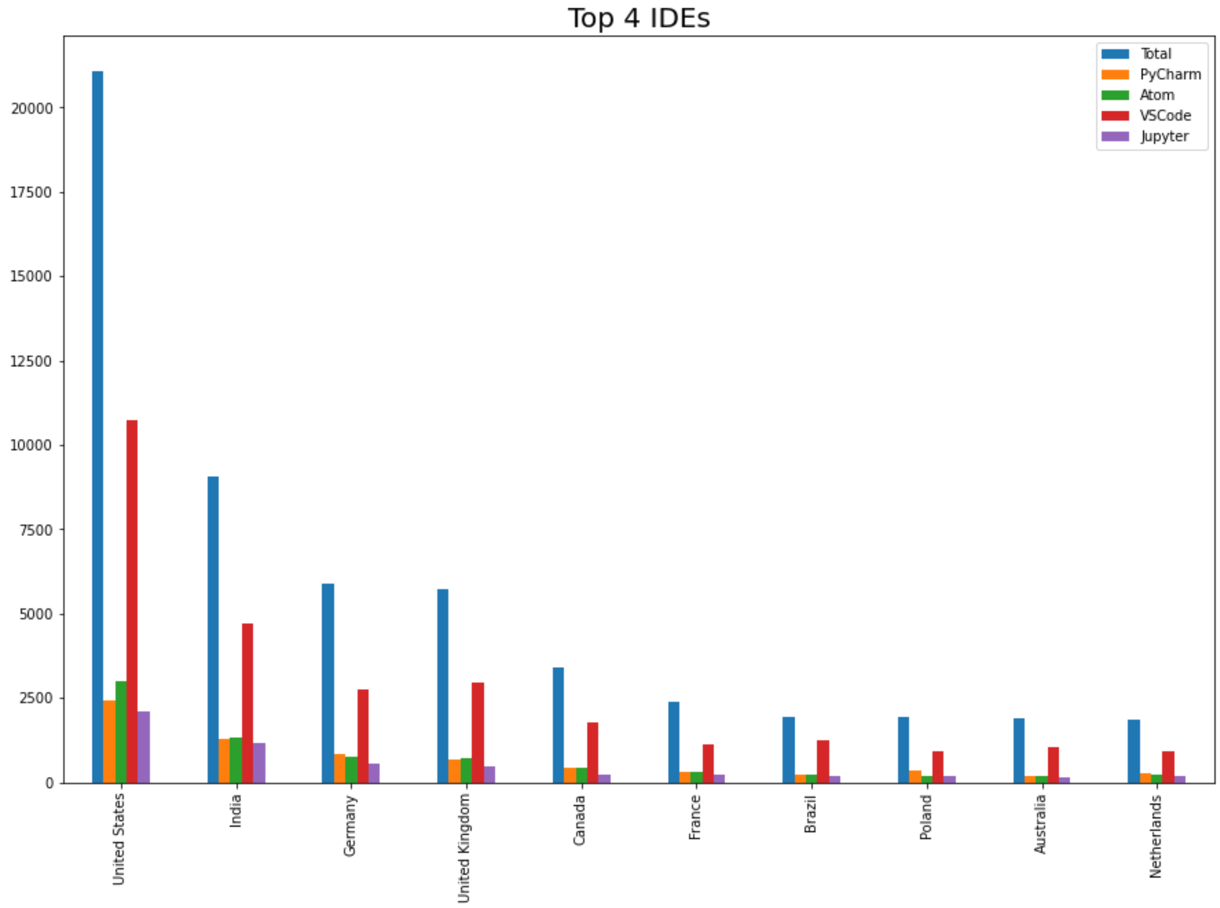
```
top_IDEs = pd.concat([respondents,pyCharm,atom,vs_code,jupyter],axis=1,keys=['Total'
```

In [427…

```
top_IDEs.head(10).plot(kind='bar',figsize=(15,10))
plt.title('Top 4 IDEs',fontsize=20)
```

Out[427…

```
Text(0.5, 1.0, 'Top 4 IDEs')
```



It seems like VS Code is most popular Development Environment across all countries.Followed by Atom and Pycharm.

In [438…

```
windows = country_grp['OpSys'].apply(lambda x : x.str.contains('Windows').sum())
Linux_based = country_grp['OpSys'].apply(lambda x : x.str.contains('Linux-based').su
MacOS = country_grp['OpSys'].apply(lambda x : x.str.contains('MacOS').sum())
```

In [439…

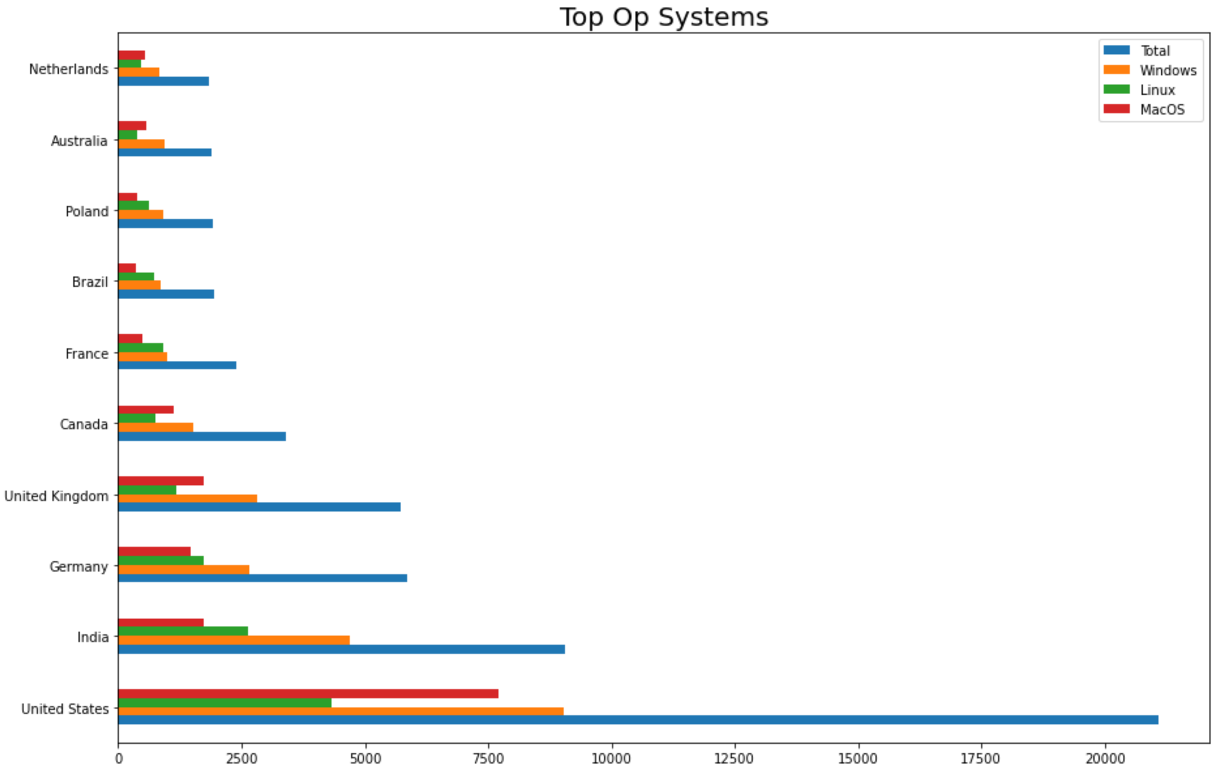
```
top_op_sys = pd.concat([respondents,windows,Linux_based,MacOS],axis=1,keys=['Total',
```

In [444…

```
top_op_sys.head(10).plot(kind='barh',figsize=(15,10))
plt.title('Top Op Systems',fontsize=20)
```

Out[444…

```
Text(0.5, 1.0, 'Top Op Systems')
```



Windows are top choice eveywhere becuase its cheap and easy to available, Followed by Mac.

In USA Mac users are very close to windows becuase its cheap there. In india Linux is second choice.

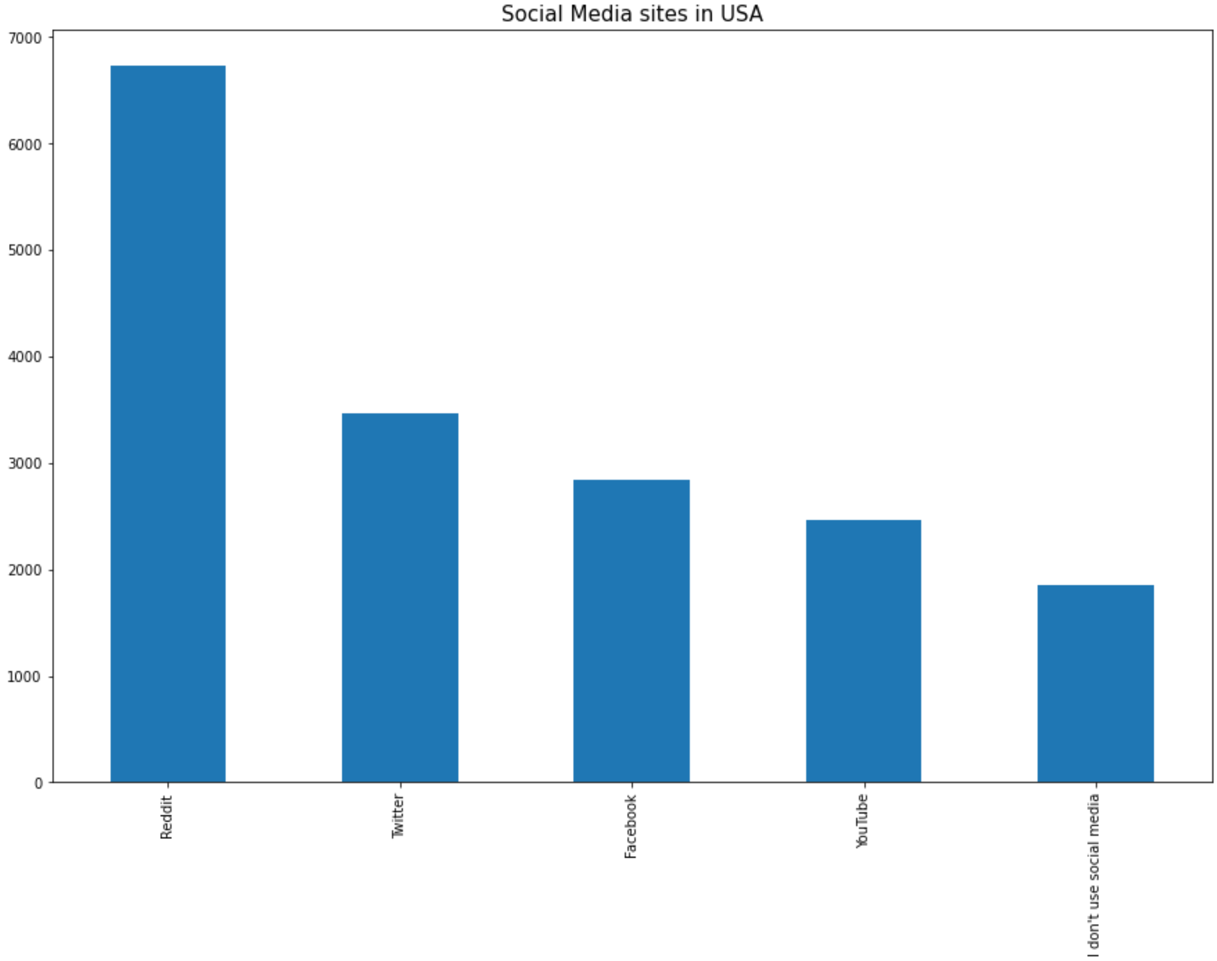
Lets find out popular social media plateform by country

In [451…

```
country_grp.get_group('United States')['SocialMedia'].value_counts().head().plot(kin
plt.title('Social Media sites in USA',fontsize=15)
```

Out[451…

```
Text(0.5, 1.0, 'Social Media sites in USA')
```

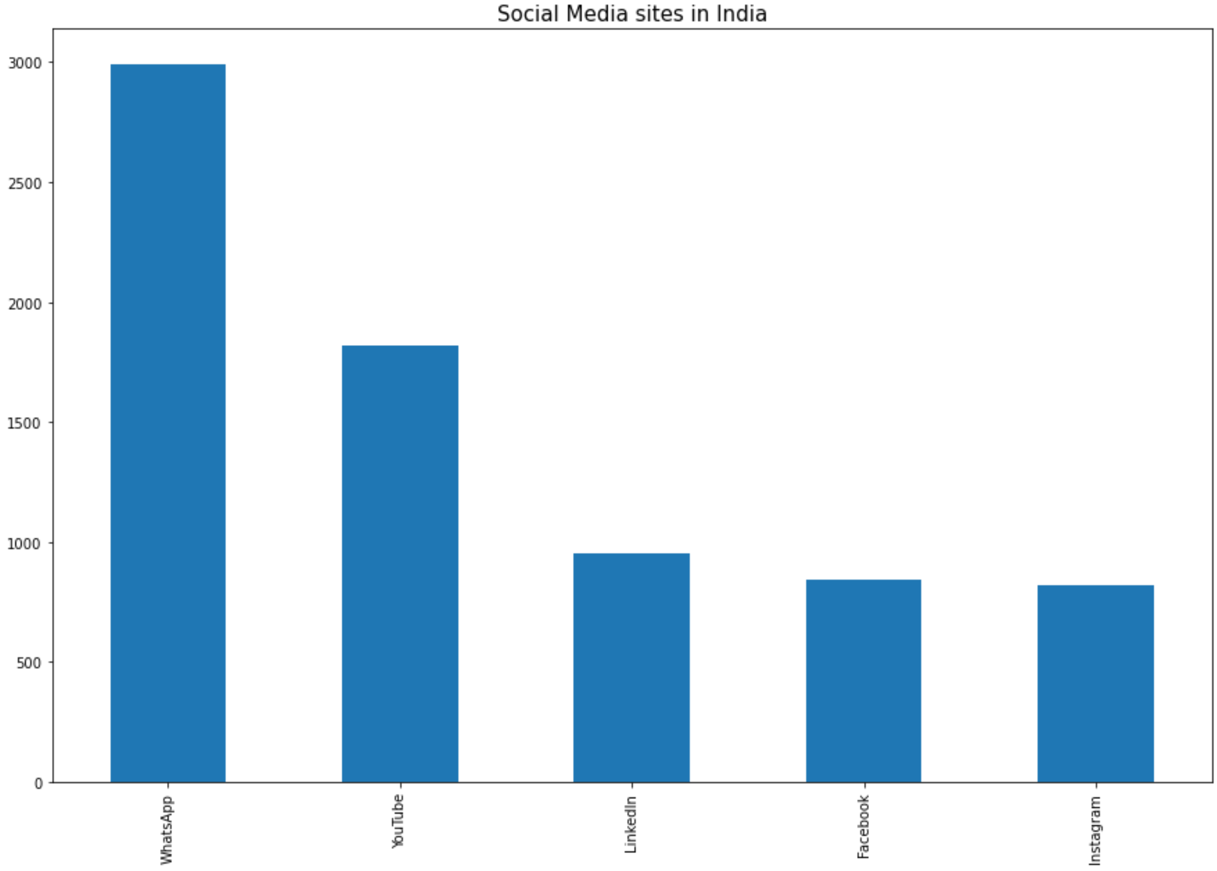


In [452…

```
country_grp.get_group('India')['SocialMedia'].value_counts().head().plot(kind='bar',
plt.title('Social Media sites in India',fontsize=15)
```

Out[452…

```
Text(0.5, 1.0, 'Social Media sites in India')
```

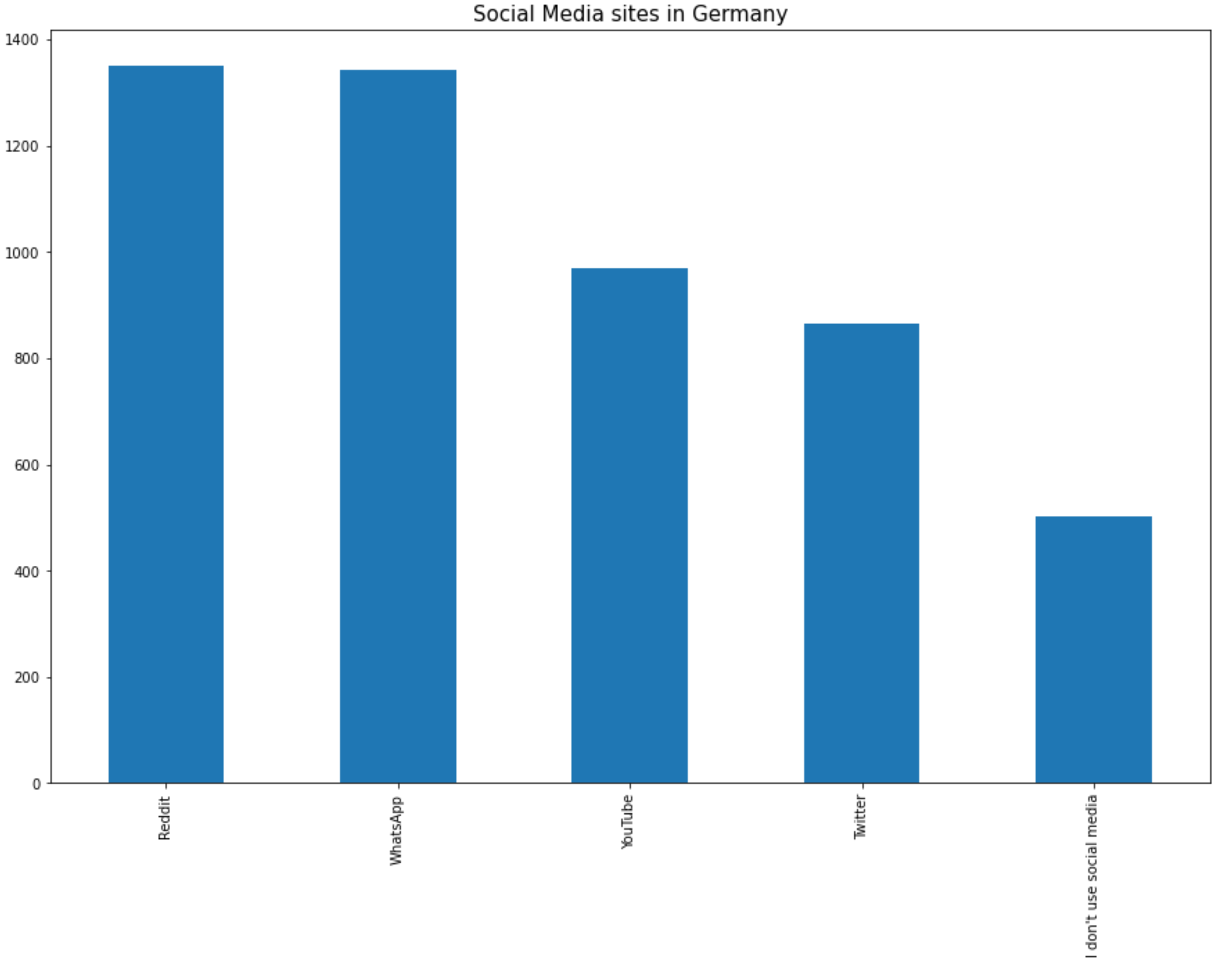


In [453…

```
country_grp.get_group('Germany')['SocialMedia'].value_counts().head().plot(kind='bar
plt.title('Social Media sites in Germany',fontsize=15)
```

Out[453…

```
Text(0.5, 1.0, 'Social Media sites in Germany')
```

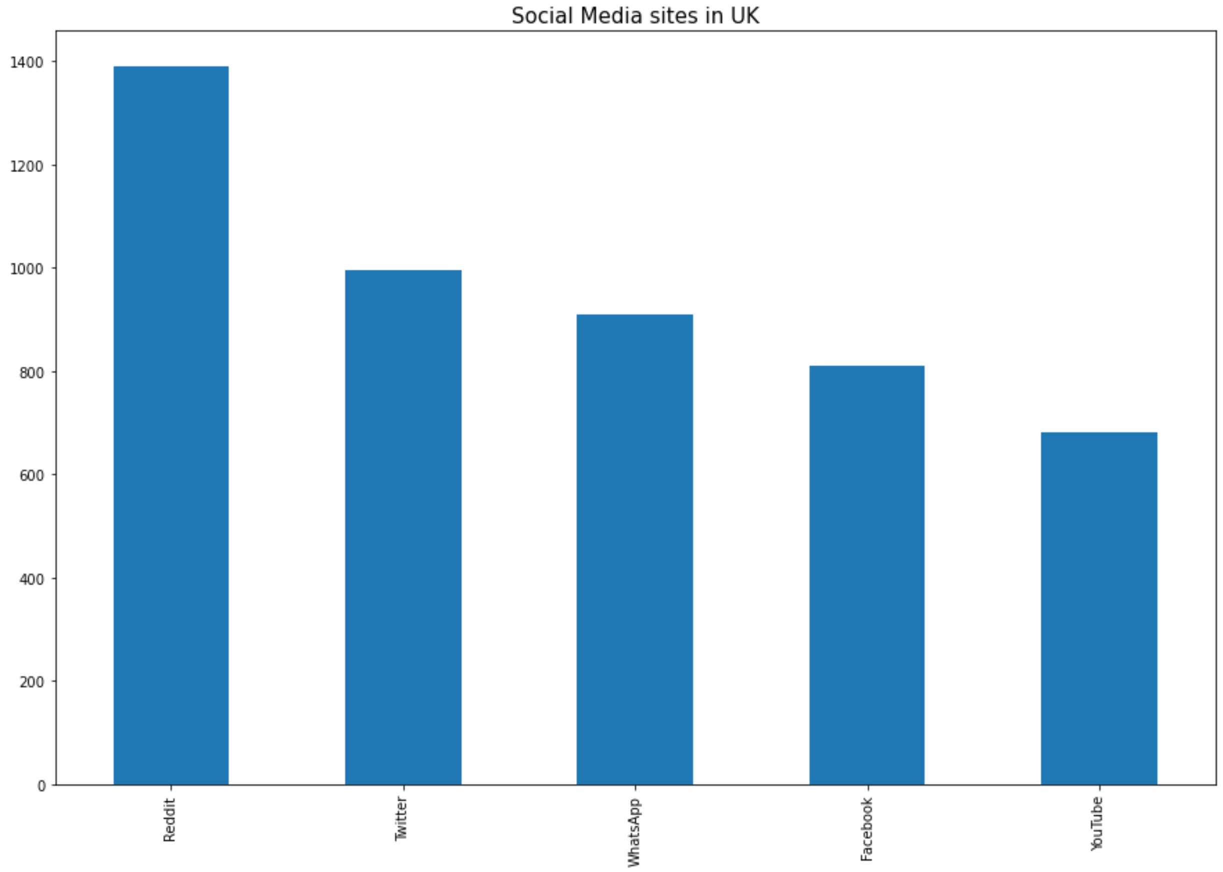


In [454…

```
country_grp.get_group('United Kingdom')['SocialMedia'].value_counts().head().plot(ki
plt.title('Social Media sites in UK',fontsize=15)
```

Out[454…

```
Text(0.5, 1.0, 'Social Media sites in UK')
```

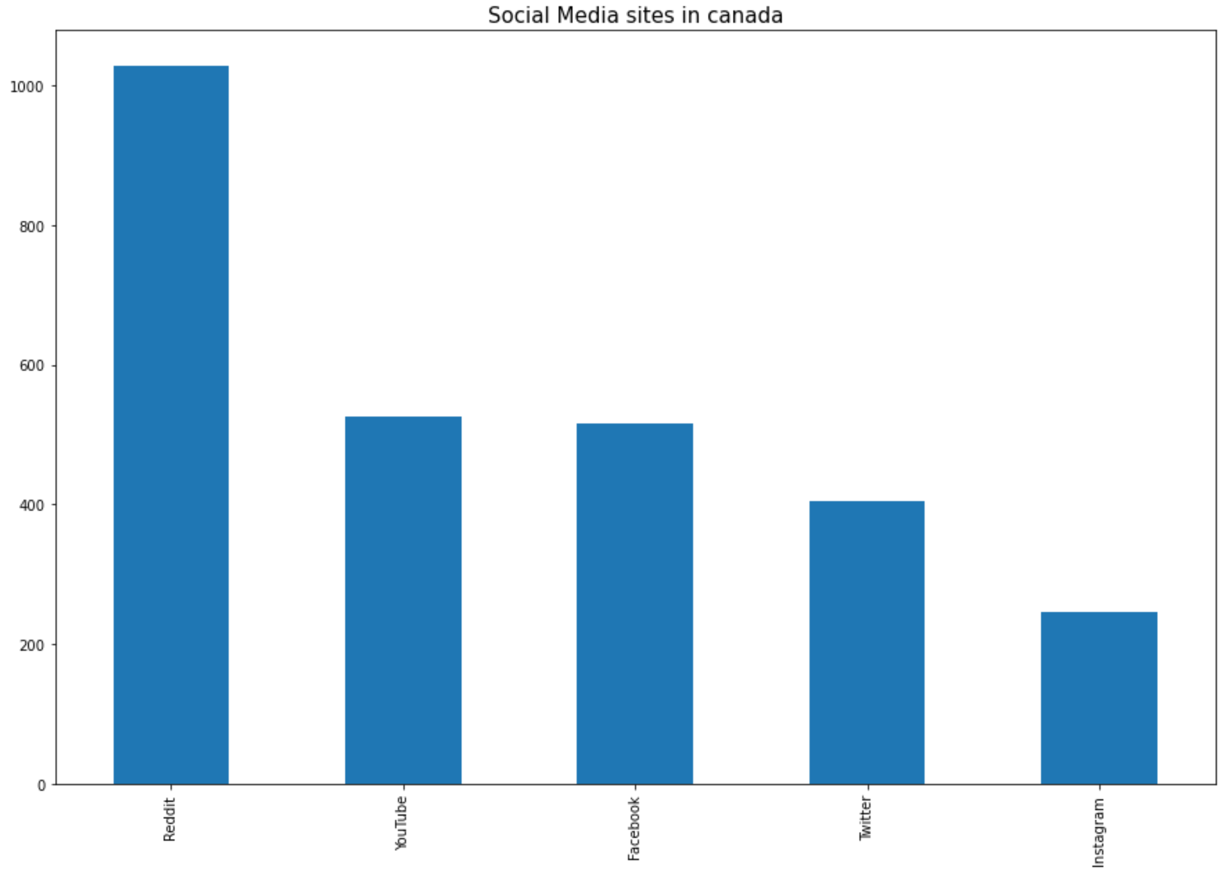


In [455…

```
country_grp.get_group('Canada')['SocialMedia'].value_counts().head().plot(kind='bar'
plt.title('Social Media sites in canada',fontsize=15)
```

Out[455…

```
Text(0.5, 1.0, 'Social Media sites in canada')
```

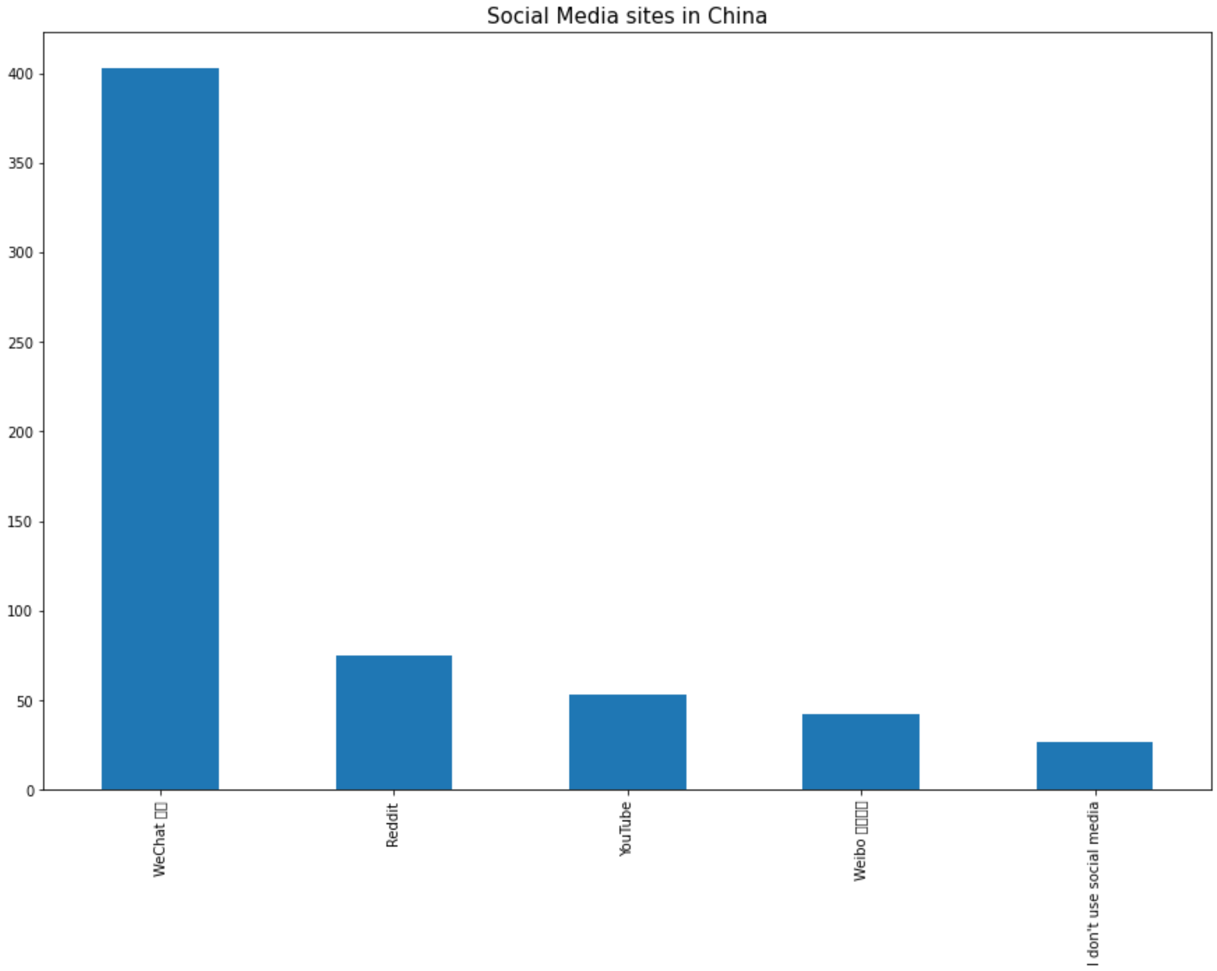


In [458…

```
country_grp.get_group('China')['SocialMedia'].value_counts().head().plot(kind='bar',
plt.title('Social Media sites in China',fontsize=15)
```

Out[458…

```
Text(0.5, 1.0, 'Social Media sites in China')
```

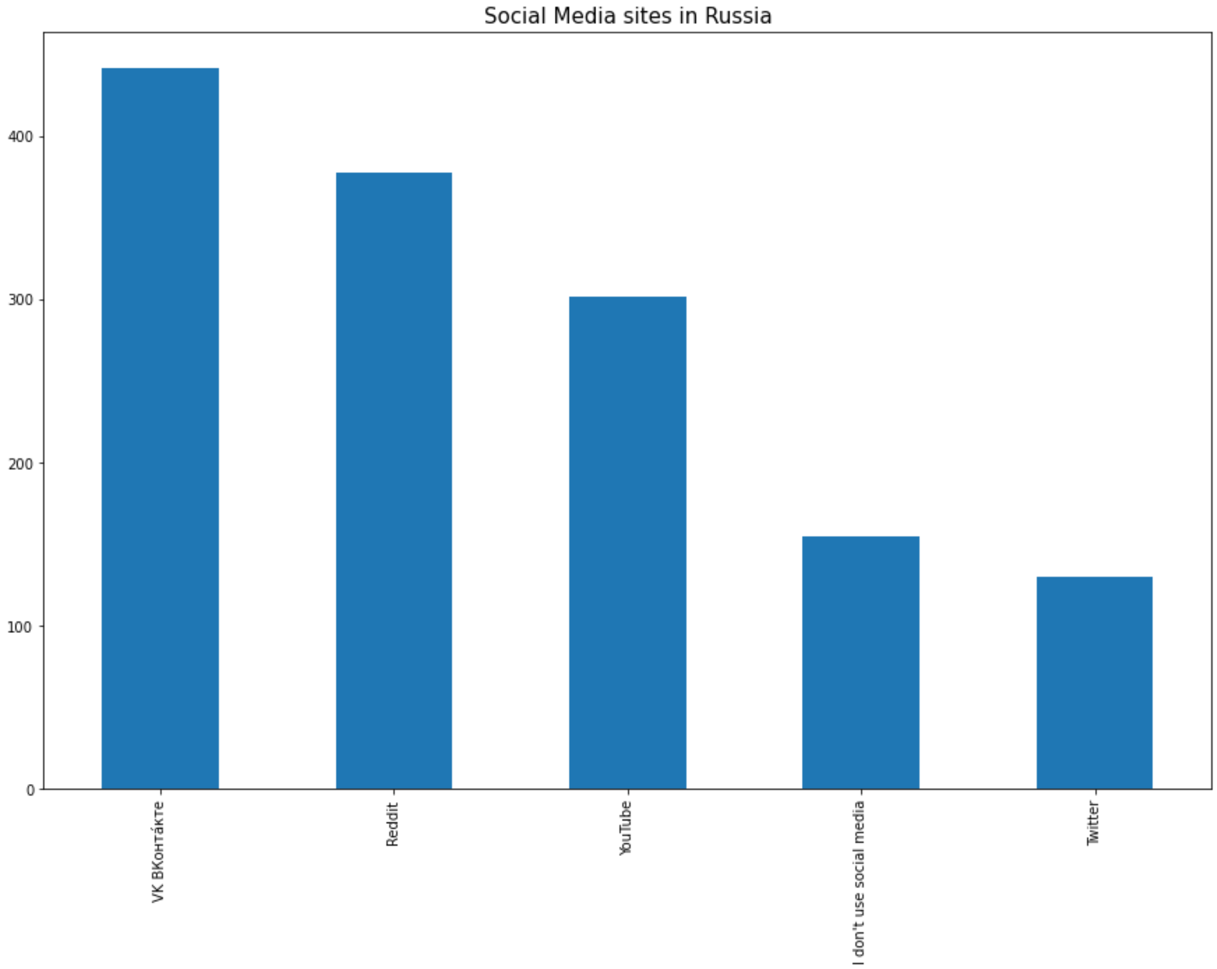


In [460…

```
country_grp.get_group('Russian Federation')['SocialMedia'].value_counts().head().plo
plt.title('Social Media sites in Russia',fontsize=15)
```

Out[460…

```
Text(0.5, 1.0, 'Social Media sites in Russia')
```



In US and Europe Reddit is most popular followed by twitter and youtube.In India whatsapp and youtube is more used.

Meanwhile China and Russia have their own Socail media plateform followed by reddit and youtube.

In [464…

```
country_grp.get_group('India')['Age'].agg(['mean','median','min','max'])
```

Out[464…

```
mean      26.522216
median    27.000000
min        1.000000
max       98.000000
Name: Age, dtype: float64
```

In [465…

```
country_grp.get_group('United States')['Age'].agg(['mean','median','min','max'])
```

Out[465…

```
mean      32.358565
median    29.000000
min        1.000000
max       99.000000
Name: Age, dtype: float64
```

In [466…

```
country_grp.get_group('China')['Age'].agg(['mean','median','min','max'])
```

Out[466…

```
mean      27.433735
median    29.000000
min        1.000000
max       70.000000
Name: Age, dtype: float64
```

In [467…

```
country_grp.get_group('Germany')['Age'].agg(['mean','median','min','max'])
```

Out[467…

```
mean      30.077105
median    29.000000
min        2.000000
max       99.000000
Name: Age, dtype: float64
```

Age feature doesn't give much information becuase of outliers but median age is almost same in every country.

In [ ]: