In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
%pylab inline

import sklearn as sk
from sklearn.cluster import KMeans
import sklearn.tree as tree
from IPython.display import Image
import pydotplus

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import warnings
warnings.simplefilter('ignore')

import plotly.offline as py
import plotly.graph_objs as go
py.init_notebook_mode(True)
from wordcloud import WordCloud, STOPWORDS
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [11]:
```
pd.set_option('display.max_columns',251)
pd.set_option('display.max_rows',251)
pd.set_option('max_colwidth',150)
```

In [2]:
```
df = pd.read_csv('HackerRank-Developer-Survey-2018-Values.csv')
```

I. Data Description:

HackerRank is platform that ensures developers and companies can find each other and best recruitment matches are made. A Survey of 35 questions, hosted on SurveyMonkey, was sent out by HackerRank to developers in late 2016 and HackerRank received responses from 25,000 developers. The survey asked professional developers many questions around their skills, educational background, current role, and more. A clean version, containing complete and non-spam responses, of the survey schema was published by HackerRank on Kaggle. Most responses were to be answered by marking the correct check box applicable to the reponse. Each response of the user was stored in seperate columns in the form of dummy columns, thus creating 250 columns within the schema.

Our team worked on the responses received from the survey to find unusual and interesting trends in the world of coding!

Data Dictionary:

Below is a short explaination to each field of the survey. We have removed the dummy columns for questions with an option to check box multiple responses.

In [58]:
```
Columndict = pd.read_csv('HackerRank-Developer-Survey-2018-Codebook (1).csv',i
ndex_col='Data Field')
Columndict.drop('Notes',axis=1)
```

Out[58]:

	Survey Question
Data Field	
RespondentID	Unique ID per responder
StartDate	When did they start (date and time)
EndDate	When did they end (date and time)
CountryNumeric2	Which Country do you belong to?
q1AgeBeginCoding	At what age did you start coding
q2Age	How old are you now?
q3Gender	What gender do you identify with?
q4Education	What is the highest level of education you have (or plan to obtain)?
q0004_other	Other (please specify)
q5DegreeFocus	What is the focus area of your degree?
q6	How did you learn how to code?
q8JobLevel	Which of the following best matches your employment level?
q8Student	Student vs. Non-student
q9CurrentRole	Which one of these best describes your current role?
q10Industry	Which best describes the industry you work in?
q12	What are the top 3 most important things you look for in a company when looking for job opportunities? (Choose 3)
q13	Based on your last job hunting experience, how did employers measure your skills?Check all that apply
q14GoodReflecAbilities	Did you feel these were a good reflection of your abilities?
q16HiringManager	Do you interview people as part of your company's hiring process?
q17	What are some of the biggest challenges you face when hiring technical talent?Check all that apply.
q18NumDevelopHireWithinNextYear	How many developers are you hiring over the next year?
q19	Which talent assessment tools are you using in your recruiting process at the first step of the interview process? Check all that apply.

Data Field	Survey Question
q20	What are the top 3 most important qualifications you look for in an engineering candidate before the onsite? Check up to 3.
q21	Which of these core competencies do you look for in software developer candidates? Check all that apply.
q22	Which of these language proficiency do you look for in software developer candidates? Check all that apply.
q25	Which Languages do you know?
q27EmergingTechSkill	Which emerging tech skill are you currently learning or looking to learn in the next year?
q30	Besides HackerRank, which other resources do you use to practice and learn coding? Check all that apply.
q32RecommendHackerRank	Would you recommend HackerRank to a friend?
q0032_other	Other (please specify)
q33HackerRankChallforJob	Have you ever taken a HackerRank challenge as part of a job interview?
q34PositiveExp	How positive was your experience?
q34IdealLengHackerRankTest	What do you think is the ideal length of a HackerRank test as part of the job interview?
q0035_other	Other (please specify)

II. Data Preparation

Cleaning Stage

1. Dropping irrelevant Columns out of the dataset

Since the dataset contains 250 columns, we removed the columns that were irrelevant to our analysis.

In [3]:
```
df.drop({'q6LearnCodeOther','q0006_other','q7Level1','q15Level2','q22LangProfGo','q22LangProfPascal','q22LangProfClojure','q22LangProfHaskell','q22LangProfLua',\
         'q22LangProfOther','q0022_other','q23FrameAngularJS','q23FrameReact','q23FrameVueDotJS','q23FrameEmber',\
         'q23FrameBackboneDotJS','q23FrameSpring','q23FrameJSF','q23FrameStruts','q23FrameNodeDotJS','q23FrameExpressJS',\
         'q23FrameMeteor','q23FrameDjango','q23FramePyramid','q23FrameRubyonRails','q23FramePadrino','q23FrameASP',\
         'q23FrameNetCore','q23FrameCocoa','q23FrameReactNative','q23FrameRubyMotion','q23FrameOther','q0023_other',\
         'q24VimorEmacs','q0024_other','q25LangGo','q25LangPascal','q25LangClojure','q25LangHaskell','q25LangLua','q25LangRust'\
         ,'q25LangTypescript','q25LangKotlin','q25LangJulia','q25LangErlang','q25LangOcaml','q25LangOther',\
         'q26FrameLearnAngularJS','q26FrameLearnReact','q26FrameLearnVueDotjs','q26FrameLearnEmber','q26FrameLearnBackboneDotjs'\
         ,'q26FrameLearnSpring','q26FrameLearnJSF','q26FrameLearnStruts','q26FrameLearnDjango','q26FrameLearnPyramid'\
         ,'q26FrameLearnRubyonRails','q26FrameLearnPadrino','q26FrameLearnASP','q26FrameLearnNetCore','q26FrameLearnNodeDotjs'\
       ,'q26FrameLearnExpressJS','q26FrameLearnMetero','q26FrameLearnCocoa','q26FrameLearnReactNative','q26FrameLearnRubyMotion'\
         ,'q26FrameLearnPadrino2','q26FrameLearnDjango2','q26FrameLearnPyramid2','q0026_other','q28LoveC','q28LoveCPlusPlus',\
    'q28LoveJava','q28LovePython','q28LoveRuby','q28LoveJavascript','q28LoveCSharp','q28LoveGo','q28LoveScala','q28LovePerl',\
      'q28LoveSwift','q28LovePascal','q28LoveClojure','q28LovePHP','q28LoveHaskell','q28LoveLua','q28LoveR','q28LoveRust',\
      'q28LoveKotlin','q28LoveTypescript','q28LoveErlang','q28LoveJulia','q28LoveOCaml','q28LoveOther','q29FrameLoveAngularJS',\
         'q29FrameLoveReact','q29FrameLoveVuedotjs','q29FrameLoveEmber','q29FrameLoveBackboneDotjs','q29FrameLoveSpring',\
         'q29FrameLoveJSF','q29FrameLoveStruts','q29FrameLoveDjango','q29FrameLovePyramid','q29FrameLoveRubyonRails',\
         'q29FrameLovePadrino','q29FrameLoveASP','q29FrameLoveNetCore','q29FrameLoveNodeDotjs','q29FrameLoveExpressJS',\
         'q29FrameLoveMeteor','q29FrameLoveCocoa','q29FrameLoveReactNative','q29FrameLoveRubyMotion','q0029_other','q31Level3',\
         'q0032_other','q36Level4'},axis=1,inplace=True)
```

2. Replacing all string '#NULL!' values in the data by NaN values

All NULL values in our dataset were represented by the string '#NULL!'. We replaced these values by NaN, so that it can be read as a null value in our Python script

In [4]:
```
df.replace(to_replace='#NULL!',value=np.nan,inplace=True)
```

3. Merging the other columns into their actual value columns

Some of the questions had a check box 'Other' for respondents who had an answer other than the options provided. These respondents were provided free text field to enter their preferred response. These responses appeared as q0000x_other, where x standns for the question number. We merged the responses in cells from the other column into the column with the question itself, thus adding a further step to the cleaning process of our data.

In [5]:
```
df.loc[df['q4Education'].isnull(),'q4Education'] = df['q0004_other']
df.loc[df['q5DegreeFocus'].isnull(),'q5DegreeFocus'] = df['q0005_other']
df.loc[df['q8JobLevel'].isnull(),'q8JobLevel'] = df['q0008_other']
df.loc[df['q9CurrentRole'].isnull(),'q9CurrentRole'] = df['q0009_other']
df.loc[df['q10Industry'].isnull(),'q10Industry'] = df['q0010_other']
df.loc[df['q27EmergingTechSkill'].isnull(),'q27EmergingTechSkill'] = df['q0027
_other']
df.drop({'q0004_other','q0005_other','q0008_other','q0009_other','q0010_other'
,'q0027_other'},axis=1,inplace=True)
```

4. Converting String data to Numerical values for analysis

For the dummy columns that were responses to the same question, we converted the string fields into 1s and 0s for better analysis

In [6]:
```
columnsList = ['q6','q12','q13','q17','q19','q20','q21','q22','q30']
for a in df.columns:
    for b in columnsList:
        if a.startswith(b):
            df[a].fillna(0,inplace=True)
            df[a] = df[a].apply(lambda x: 0 if x==0 else 1)
```

In [7]:
```
for r in df.columns:
    if r.startswith('q25'):
        df[r] = df[r].apply(lambda x: 0 if x=='Will Learn' else (1 if x=='Kno
w' else 0))
```

In [8]:
```
df['q32RecommendHackerRank']=df['q32RecommendHackerRank'].apply(lambda x: 0 if
 x=='No' else (1 if x=='Yes' else np.nan))

df['q33HackerRankChallforJob']=df['q33HackerRankChallforJob'].apply(lambda x:
0 if x=='No' \
                                                                    else
 (1 if x=='Yes' else np.nan))
```

5. Adding extra colums

We added an extra column 'isNonEngineer' to mark respondents who have a degree focus apart from Engineering or STEM. This column is used later during our analysis for insight 1.

In [9]:
```
df['isNonEngineer'] = df['q5DegreeFocus'].apply(lambda x: 0 if x=='Computer Sc
ience' else\
                                                0 if x=='Other STEM (science,
 technology, engineering, math)' else 1)
```

III. General Analysis

1. Survey Responses Around the World

Below is a map of the world showing response rate of developers around the world. It can be seen that the maximum respondents were from India and United States, followed by Canada, Brazil and United Kingdom.

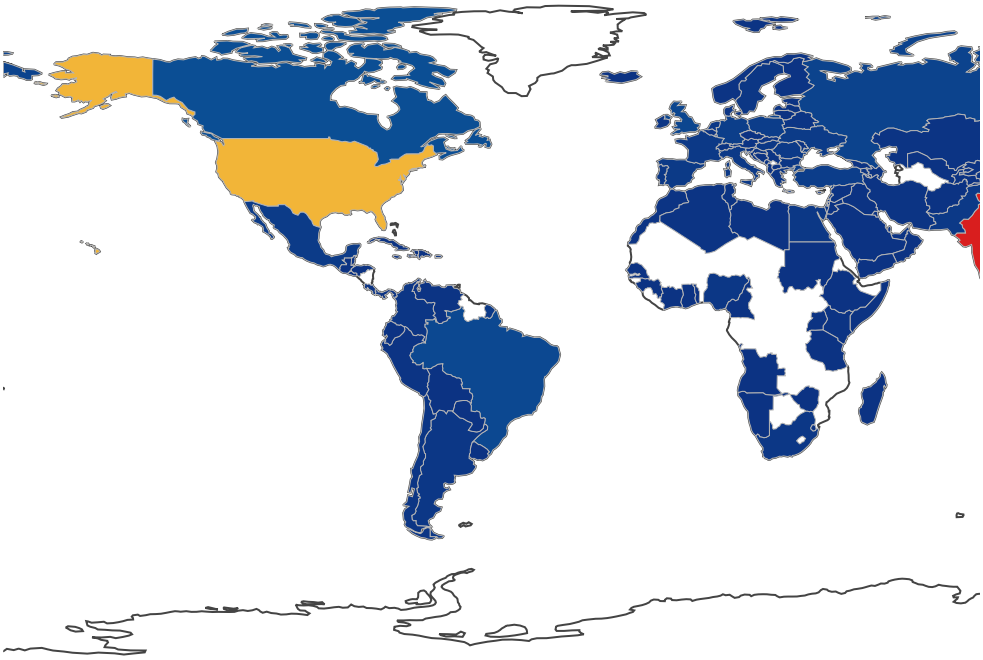
In [3]:
```
# Count by country

poo = df['CountryNumeric2'].value_counts()

# plotly
data = [dict(
    type ='choropleth',
    locations = poo.index,
    locationmode ='country names',
    z = poo.values,
    text = ('Count'+'<br>'),
    colorscale='Portland',
    reversescale=False,
    marker=dict(line=dict(color='rgb(180,180,180)', width=0.5)),

    colorbar = dict(title ='Response count')
    )]
layout = dict(title = 'Number of response by country',
             geo = dict( showframe= False,
                       showcoastlines =True,
                       projection = dict(type = 'Mercator')))
fig = dict(data=data, layout=layout)
py.iplot(fig)
```

Number of response by coun



Reference : https://www.kaggle.com/sudhirnl7/hacker-rank-qna-eda (https://www.kaggle.com/sudhirnl7/hacker-rank-qna-eda)

Here is the list of top 10 countries with the most number of respondents.

In [67]:
```
df.groupby('CountryNumeric2')['RespondentID'].count().nlargest(10).reset_index()
```

Out[67]:

	CountryNumeric2	RespondentID
0	India	8088
1	United States	4937
2	Canada	642
3	Brazil	502
4	United Kingdom	443
5	Indonesia	387
6	Russian Federation	378
7	Germany	258
8	Turkey	250
9	Poland	248

2. Gender of Respondents

Here is a count of number of Male and Female survey respondents

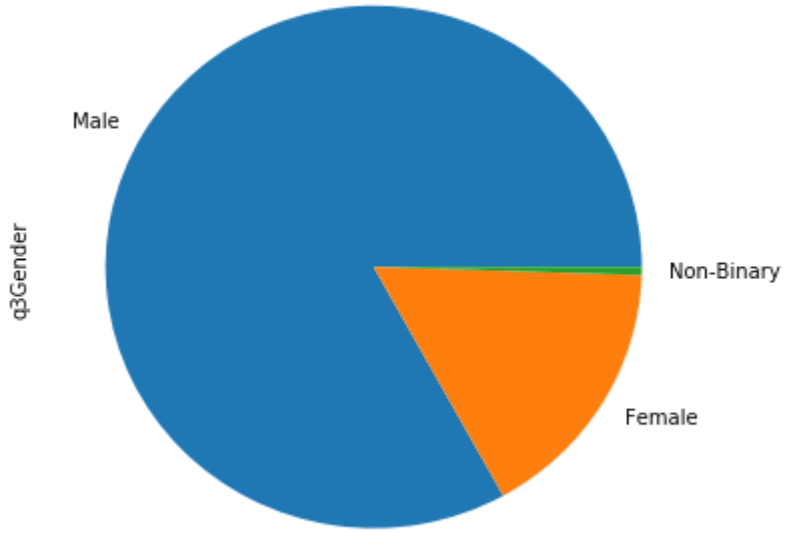
In [68]:
```
df.groupby('q3Gender')['RespondentID'].count().reset_index()
```

Out[68]:

	q3Gender	RespondentID
0	Female	4122
1	Male	20774
2	Non-Binary	125

In [69]:
```
df.q3Gender.value_counts().plot.pie(figsize=(6, 6))
```

Out[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x4abea302e8>
```



3. Job Roles of Respondents

Here is a list of the top 15 most common roles of the survey respondents

In [70]:
```
df.groupby('q9CurrentRole')['RespondentID'].count().nlargest(15).reset_index()
```

Out[70]:

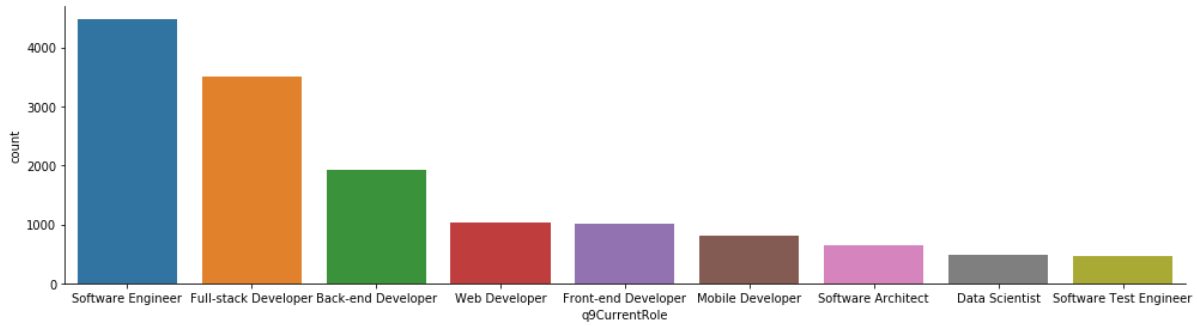
	q9CurrentRole	RespondentID
0	Student	8006
1	Software Engineer	4482
2	Full-stack Developer	3498
3	Back-end Developer	1922
4	Web Developer	1032
5	Front-end Developer	1006
6	Mobile Developer	812
7	Software Architect	643
8	Data Scientist	485
9	Software Test Engineer	454
10	Unemployed	383
11	Data Analyst	349
12	Development Operations Engineer	251
13	Data Engineer	216
14	Software Specialist	148

Represents the top roles with Student not a part of this. Only professional roles are visualized

In [71]:
```
dfRoles = df[df['q9CurrentRole']!='Student'].groupby('q9CurrentRole')['Respond
entID'].count()\
.nlargest(9).reset_index(name='count')
```

In [72]:
```
sns.factorplot(x='q9CurrentRole',y='count',data=dfRoles,kind='bar',aspect=3.5)
```

Out[72]: `<seaborn.axisgrid.FacetGrid at 0x4abe2bbe80>`



4. Future requirement of professionals

Here we are trying to present the requirement for skilled developers in the next one year.

In [73]:
```
df[df.q16HiringManager=='Yes'].groupby('q18NumDevelopHireWithinNextYear')['RespondentID'].count().sort_values()
```

Out[73]:
```
q18NumDevelopHireWithinNextYear
501 - 1000      45
1000+           90
201 - 500       96
101 - 200      174
51 - 100       350
11 - 50       1501
0 - 10        5555
Name: RespondentID, dtype: int64
```

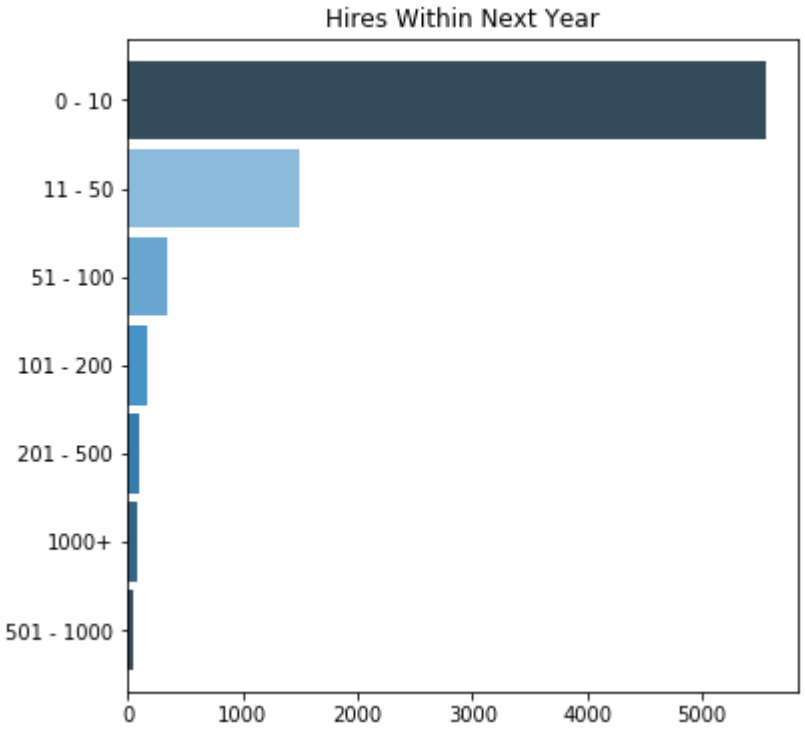
In [74]:
```
dfDev = df[df.q16HiringManager=='Yes'].groupby('q18NumDevelopHireWithinNextYear')['RespondentID'].count().sort_values().reset_index(name='count')
```

In [75]:
```
plt.figure(figsize=(6,6))
df.q18NumDevelopHireWithinNextYear.value_counts().sort_values(ascending=True).
plot\
.barh(width=0.9,color=sns.color_palette('Blues_d'))
plt.title('Hires Within Next Year')
plt.show()
```

Out[75]: <matplotlib.figure.Figure at 0x4ac4bcbeb8>

Out[75]: <matplotlib.axes._subplots.AxesSubplot at 0x4ac55a54e0>

Out[75]: Text(0.5,1,'Hires Within Next Year')



5. Emerging Tech Skills

These are the top 10 emerging skills according to the responses of HackerRank users who took this survey.

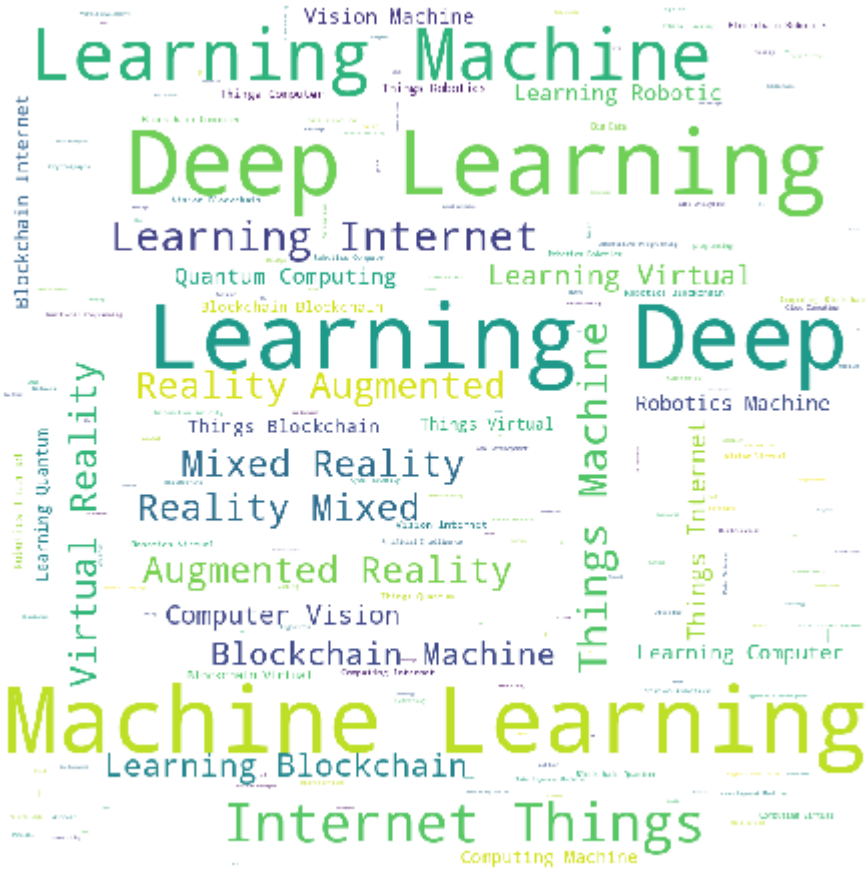
In [80]:
```
plt.figure(figsize=(16,8))
wc = WordCloud(background_color="white", max_words=1000,
               stopwords=STOPWORDS,width=1000,height=1000)
wc.generate(" ".join(df['q27EmergingTechSkill'].dropna()))
plt.imshow(wc)
plt.axis('off')
plt.show()
```

Out[80]: <matplotlib.figure.Figure at 0x4ac5483390>

Out[80]: <wordcloud.wordcloud.WordCloud at 0x4acb2cc048>

Out[80]: <matplotlib.image.AxesImage at 0x4ac5469320>

Out[80]: (-0.5, 999.5, 999.5, -0.5)



IV. Insight 1: Trends for reponsdents having a degree focus apart from Computer Science or STEM

Our first Insight looks at the trends for respondents who had a degree focus apart from Computer Science or STEM. We chose this particular cluster of people to understand and prove that coding is a skill that can be acheived without having a strong foundation in computer science via a degree.

1. Create a series containing top 15 Non Engineering degrees

In [81]:
```
nonEnggDegrees = df[~df.q5DegreeFocus.isin(['Computer Science','Other STEM (sc
ience, technology, engineering, math)'])]\
.groupby('q5DegreeFocus')['RespondentID'].count().nlargest(15)
```

2. Create a series containing people in top 10 Roles who are from Non Engineering background

In [82]:
```
topRoles = df[df['q5DegreeFocus'].isin(nonEnggDegrees.index)].groupby(['q9Curr
entRole']).size().nlargest(10)
```

3. Create a series of top 5 countries based on responses recieved

In [44]:
```
topCoun = df.groupby(['CountryNumeric2']).size().nlargest(5)
```

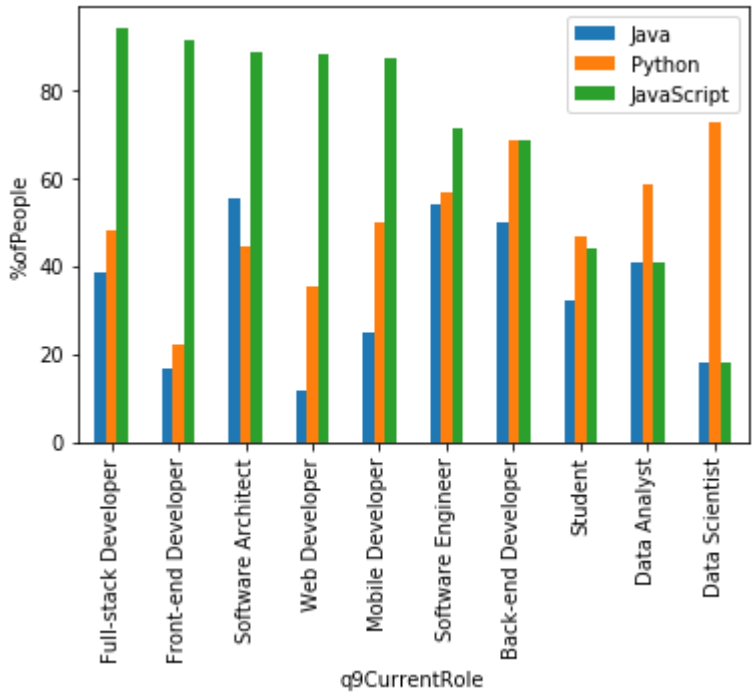
4. Average programming language skills of all people belonging to Non Engineering background grouped by their Current Job level

In [84]:
```
df1 = df[(df['q5DegreeFocus'].isin(nonEnggDegrees.index))&(df['q9CurrentRole']
.isin(topRoles.index))].groupby(['q9CurrentRole'])\
['q25LangC','q25LangCPlusPlus','q25LangJava','q25LangPython',\
'q25LangRuby','q25LangJavascript','q25LangCSharp','q25LangPHP','q25LangR'].mea
n()*100

df1 = df1.reset_index().rename(columns={'q25LangJava':'Java','q25LangPython':
'Python','q25LangJavascript':'JavaScript'})\
.sort_values(by=['JavaScript','Java'],ascending=False)
```

In [85]:
```
plot1 = df1.plot(x="q9CurrentRole", y=["Java", "Python", "JavaScript"], kind="bar")
plot1.set_ylabel("%ofPeople")
```

Out[85]: Text(0,0.5,'%ofPeople')



As per our team's perception, we believed that people from a non-technical background would prefer to work as a Data Scientist or Data/Business Analyst. But, from our analysis, it can be seen that developers with a degree focus apart from Computer Science or STEM are now working as Developers and are highly proficient in JavaScript and Java.

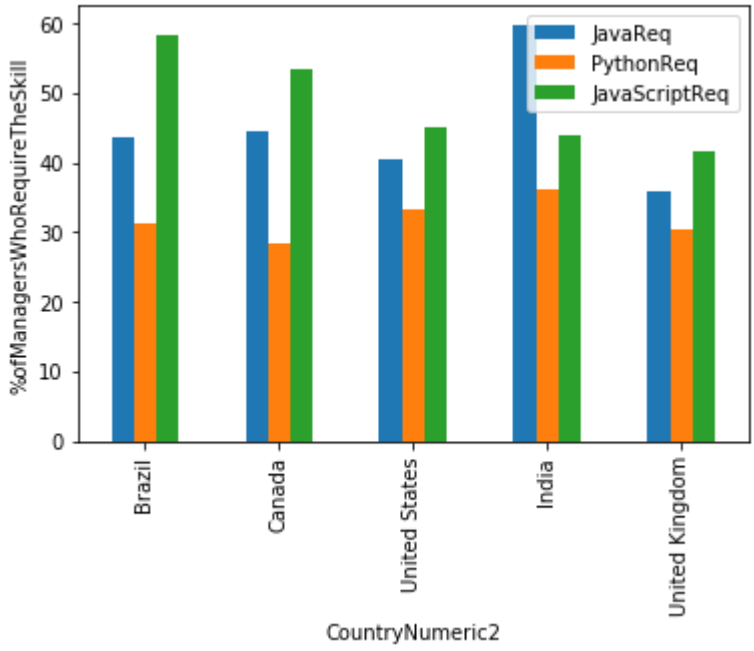
5. Countrywise programming language proficiency required by Hiring Managers

In [86]:
```
df2=df[(df['q16HiringManager']=='Yes')&(df['CountryNumeric2'].isin(topCoun.index))]\
.groupby(['CountryNumeric2'])['q22LangProfC','q22LangProfCPlusPlus','q22LangProfJava','q22LangProfPython',\
 'q22LangProfRuby', 'q22LangProfJavascript','q22LangProfCSharp','q22LangProfPHP','q22LangProfR'].mean()*100

df2 = df2.reset_index().rename(columns={'q22LangProfJava':'JavaReq','q22LangProfPython':'PythonReq',\
                                        'q22LangProfJavascript':'JavaScriptReq'}).sort_values(by='JavaScriptReq',ascending=False)
```

In [87]:
```
plot2 = df2.plot(x="CountryNumeric2", y=["JavaReq", "PythonReq", "JavaScriptRe
q"], kind="bar")
plot2.set_ylabel("%ofManagersWhoRequireTheSkill")
```

Out[87]: Text(0,0.5,'%ofManagersWhoRequireTheSkill')



We have backed up our insight by showing that hiring managers from the top 5 respondent countries require proficiency in JavaScript, which could be the reason why developers from non-engineering degree focus prefer working with JavaScript

Managerial Insights:

Using our analysis, it can be derived that Hiring Managers should widen their scope while hiring for developer roles by focusing on non-engineering backgroud candidates as well, and not just limiting their search within computer science and STEM.

Machine Learning Mastery:

Supporting Insight 1 by using K-Means Clustering for our data

We used newly added column 'isNonEngineer' for our machine learning analysis for Insight 1.

Create another dataframe with age, education information and language skills of Non Engineering background people

In [19]:
```
dfClus = df[['q14GoodReflecAbilities','q2Age','isNonEngineer','q4Education','q
25LangC', 'q25LangCPlusPlus', 'q25LangJava','q25LangPython', \
                     'q25LangRuby', 'q25LangJavascript', 'q25LangCSharp','q25Sc
ala', 'q25LangPerl', 'q25LangSwift',\
                     'q25LangPHP', 'q25LangR']]
```

Assign numeric values to selected Age groups and Education levels

In [20]:
```
def fnAge(x):
    if 'Under 12 years old' in x:
        return 6
    elif '12 - 18 years old' in x:
        return 15
    elif '18 - 24 years old' in x:
        return 21
    elif '25 - 34 years old' in x:
        return 30
    elif '35 - 44 years old' in x:
        return 40
    elif '45 - 54 years old' in x:
        return 50
    elif '55 - 64 years old' in x:
        return 60
    elif '65 - 74 years old' in x:
        return 70

def fnCollege(x):
    if 'Some college' in x:
        return 3
    elif 'College graduate' in x:
        return 3
    elif 'Post graduate degree (Masters, PhD)' in x:
        return 4
    elif 'Some post graduate work (Masters, PhD)' in x:
        return 4
    elif 'High school graduate' in x:
        return 2
    elif 'Some high school' in x:
        return 2
    elif 'Vocational training (like bootcamp)' in x:
        return 1
```

In [21]:
```
dfClus['q2Age'] = dfClus['q2Age'].astype(str).apply(fnAge)
dfClus['q4Education'] = dfClus['q4Education'].astype(str).apply(fnCollege)
dfClus['q14GoodReflecAbilities'] = dfClus['q14GoodReflecAbilities'].apply(lamb
da x:5 if x=='Very Good' \
                        else 4 if x=='Good' else 3 if x=='Acceptable' else 2 i
f x=='Poor' else 1 if x=='Very Poor' else 0)
```

In [22]:
```
dfClus.head()
```

Out[22]:

	q14GoodReflecAbilities	q2Age	isNonEngineer	q4Education	q25LangC	q25LangCPlu
0	3	21.0	0	3.0	0	0
1	4	30.0	0	4.0	0	0
2	3	15.0	0	3.0	0	0
3	3	15.0	0	3.0	0	1
4	4	30.0	1	3.0	0	0

Remove rows containing null values for Age and Education Column

In [23]:
```
dfClus=dfClus[~dfClus.q2Age.isnull()]
dfClus=dfClus[~dfClus.q4Education.isnull()]
```

Divide the data into three clusters

In [24]:
```
clu = KMeans(n_clusters=3, random_state=0)
clu
clu.fit(dfClus)
clu.labels_
```

Out[24]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=0, tol=0.0001, verbose=0)
```

Out[24]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=0, tol=0.0001, verbose=0)
```

Out[24]:
```
array([0, 1, 0, ..., 0, 0, 0])
```

In [25]:
```
dfClus['cluster']=clu.labels_
```

In [26]:
```
dfClus.groupby('cluster').mean()
```

Out[26]:

	q14GoodReflecAbilities	q2Age	isNonEngineer	q4Education	q25LangC	q25
cluster						
0	3.442676	20.545964	0.024066	3.162930	0.772347	0.67
1	3.531350	30.000000	0.076042	3.377258	0.568072	0.50
2	3.455234	42.413912	0.105028	3.427342	0.619835	0.55

As we can see, cluster 2 contains the maximum percentage of non-engineers and it also contains the maximum percentage for proficiency in JavaScript amongst the other 2 clusters

V. Insight 2: Women Power

Our second insight looks at interesting trends for women in technology. We chose this particular cluster for our analysis to see women's contribution in the technology field

1. Finding % population for both genders in their respective AgeWhenBeginCoding brackets

In [11]:
```
GenderAnalysis = (df[df.q3Gender!='Non-Binary'].groupby(['q1AgeBeginCoding','q
3Gender']).size()/df.groupby('q3Gender').size()).reset_index()
```

Replace label to allow proper sorting on the column AgeWhenBeginCoding

In [12]:
```
GenderAnalysis.replace(to_replace='5 - 10 years old',value='05 - 10 years old'
,inplace=True)
```

2. Convert the ratios into Percentages

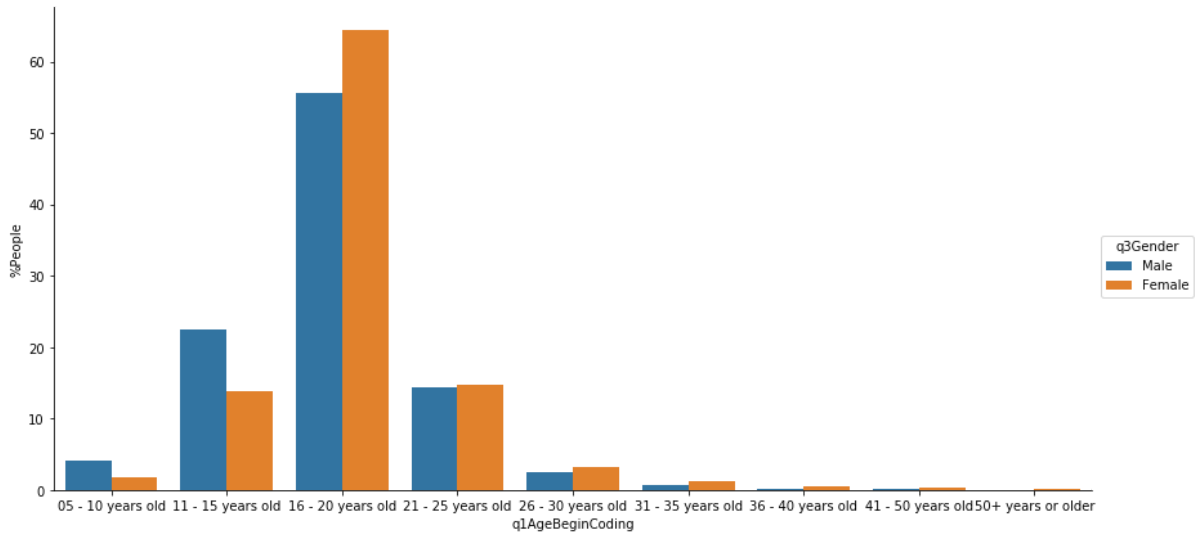
In [13]:
```
GenderAnalysis[0] = GenderAnalysis[0]*100
GenderAnalysis.rename(columns={0:'%People'},inplace=True)
```

3. Sort values by Age when they began coding

In [14]:
```
womenpower = GenderAnalysis.sort_values(by='q1AgeBeginCoding')
```

In [15]:
```
sns.factorplot(x='q1AgeBeginCoding',y='%People',hue='q3Gender',kind="bar",data
=womenpower,size=6,aspect=2)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x1de150d8128>



From the graph, we can see that, apart from the first two age buckets, the percentage of women who start coding is more in every age bracket. This shows that more women begin coding early and advance their careers to pursue developer jobs.

To back this up, we created a graph as shown below.

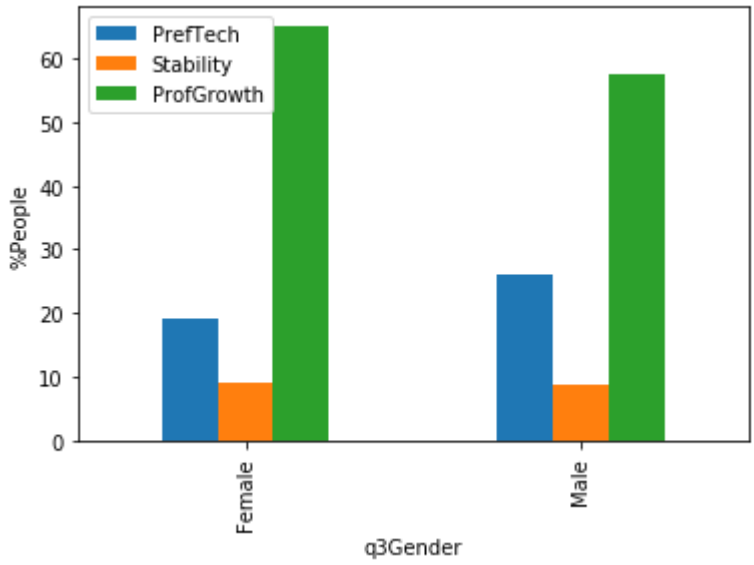
4. Prefered technology stack, Stability and Professional growth choices of people from Tech Industry and belonging to top 5 countries, segregated by gender

In [45]:
```
df3 = df[(df.CountryNumeric2.isin(topCoun.index)&(df.q10Industry=='Technology'
))&(df.q3Gender!='Non-Binary')].groupby(['q3Gender'])\
['q12JobCritPrefTechStack','q12JobCritStability','q12JobCritProfGrowth'].mean
()*100
```

In [46]:
```
df3 = df3.reset_index().rename(columns={'q12JobCritPrefTechStack':'PrefTech',
'q12JobCritStability':'Stability'\
                                        ,'q12JobCritProfGrowth':'ProfGrowth'})
```

In [47]:
```
plot3 = df3.plot(x='q3Gender',y=['PrefTech','Stability','ProfGrowth'],kind='ba
r')
plot3.set_ylabel("%People")
```

Out[47]: Text(0,0.5,'%People')



It can also be seen that while looking for a job, Women value Professional Growth more and are less comfortable working with their known technology stack, when compared to Men. Which shows that women are more open to learning new programming languages during their careers. Another interesting finding that we noticed was that both Males and Females do not give a lot of preference to Job Stability, which is contrary to popular belief.

Managerial Insights:

From our analysis, it can be derived that women are open to working with technologies outside of their experience to improve their professional growth. Hiring Managers can use this insight while recruiting women candidates and venture out to a larger pool of candidates.

Machine Learning Mastery:

We used a predictor to predict if women with certain skills and age groups will end up with a technical job

In [27]:
```
Predictor = dfClus.copy()
```

1. Taking extra columns from main dataframe as per requirement

In [28]:
```
Predictor['q2Age']=df['q2Age']
Predictor['q1AgeBeginCoding']=df['q1AgeBeginCoding']
Predictor['q3Gender']=df['q3Gender']
Predictor['q9CurrentRole']=df['q9CurrentRole']
```

2. Making values of all columns as numeric to do clustering on the data

In [29]:
```
Predictor=Predictor[Predictor['q3Gender']!='Non-Binary']
Predictor.drop(['q14GoodReflecAbilities'],axis=1, inplace=True)
Predictor['q3Gender'] = Predictor['q3Gender'].apply(lambda x: 0 if x=='Male' e
lse 1)
Predictor['q9CurrentRole']=Predictor['q9CurrentRole'].apply(lambda x: 0 if x==
'Student' else 0 if x=='Unemployed' else 1)
Predictor = pd.get_dummies(Predictor, columns=['q2Age','q1AgeBeginCoding'])
```

3. Setting our decision variable to CurrentRole to check if person is Employed or Unemployed

In [30]:
```
X = Predictor.drop('q9CurrentRole',axis=1)
Y = Predictor.q9CurrentRole
```

In [31]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.1,random_s
tate = 0)
```

In [32]:
```
from sklearn.neural_network import MLPClassifier
cl = MLPClassifier()
cl.fit(X_train,Y_train)
```

Out[32]:
```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(100,), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,
       verbose=False, warm_start=False)
```

In [33]:
```
y_pred = cl.predict(X_test)
```

4. Accuracy:

In [34]:
```
(y_pred == Y_test).mean()
```

Out[34]:
```
0.7869318181818182
```

In [35]:
```
y_pred_proba = cl.predict_proba(X_test)[:,1]
```

5. Precision:

In [36]:
```
import sklearn.metrics as met
```

In [37]:
```
met.precision_score(Y_test,y_pred)
```

Out[37]:
```
0.8242530755711776
```

6. Recall

In [38]:
```
met.recall_score(Y_test,y_pred)
```

Out[38]:
```
0.8621323529411765
```

7. AUC score

In [39]:
```
met.roc_auc_score(Y_test,y_pred_proba)
```

Out[39]:
```
0.8671731387867646
```

8. Running the predictor for a test data created

In [40]:
```
dftest = pd.read_csv('test data.csv')
```

Expected output: [1,1,1,1,1,1,0,0]

In [41]:
```
expected = [1,1,1,1,1,1,0,0]
```

In [42]:
```
testout = cl.predict(dftest)
```

9. Accuracy check:

In [43]:
```
(expected==testout).mean()
```

Out[43]: 0.875

We fed training data into our predictor from our dataset. Upon running the test data, it was seen that the predictor was able to predict the right result with 87.5%-100% accuracy based on multiple test runs. Thus, it is evident that women with proficiency in programming languages have high chances of being employed.

VI. Insight 3: Value of GitHub and OpenSource Contribution in different industries

The third insight looks at the trends of Hiring Managers in different industries and how much they value GitHub Projects and OpenSource Contribution of candidates.

1. Top 10 Industries based on responses

In [121]:
```
topInd = df.groupby('q10Industry')['RespondentID'].count().nlargest(10)
```

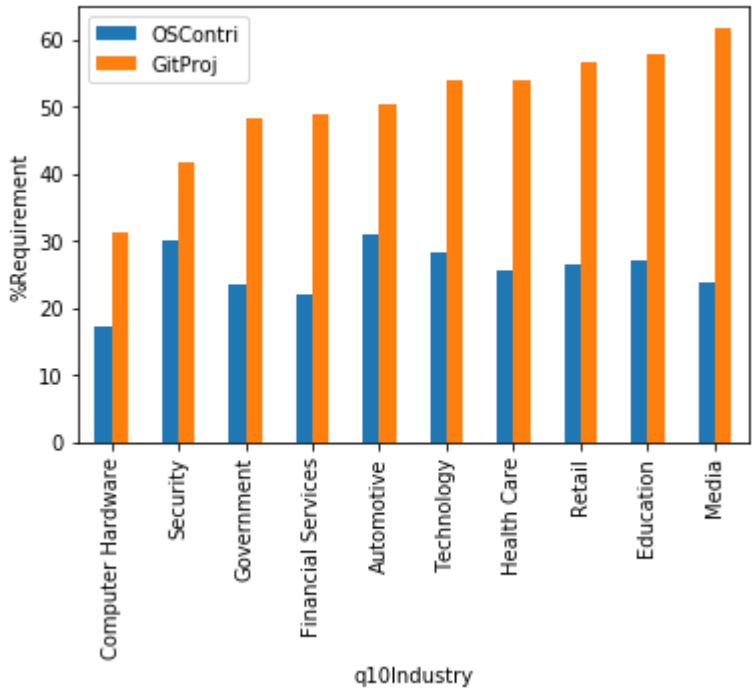
2. Parameters Hiring managers prefer while Hiring in different industries in top 5 countries

In [122]:
```
industrywiseGit = df[(df['CountryNumeric2'].isin(topCoun.index)) & (df['q10Ind
ustry'].isin(topInd.index)) &
                     (df['q16HiringManager'] == 'Yes') ]\
.groupby(['q10Industry'])['q20CandYearExp','q20CandCompScienceDegree','q20Cand
HackerRankActivity','q20CandGithubPersProj2'\
,'q20CandOpenSourceContrib','q20CandHackathonPart','q20CandPrevWorkExp'].mean
()*100

industrywise = industrywiseGit.reset_index().sort_values(by='q20CandGithubPers
Proj2').replace(to_replace='Automotive & Transportation',value='Automotive').\
                rename(columns={'q20CandOpenSourceContrib':'OSContri','q20Cand
GithubPersProj2':'GitProj'})#.sort_values(by={'GitProj'})
```

In [123]:
```
plot4 = industrywise.plot(x='q10Industry',y=['OSContri','GitProj'],kind='bar')
plot4.set_ylabel("%Requirement")
```

Out[123]: Text(0,0.5,'%Requirement')



It can be seen that Hiring Managers from non-Technology Industries consider candidate's GitHub Projects and OpenSource Contributions as an important qualification while hiring. This could be because Hiring Managers from non-technology industries have limited resources in their IT team and would want to hire only the most proficient and experienced candidates who have worked on several projects

Managerial Insights:

Along with other factors, more hiring managers should give significant importance to a candidate's GitHub projects and OpenSource contributions.

Machine Learning Mastery:

We created a decision tree using columns which are important for this analysis

We consider columns which show what hiring managers value while hiring

In [124]:
```
dfManagers = df[df['q16HiringManager']=='Yes'][['q3Gender','q10Industry','q20C
andHackerRankActivity',\
 'q20CandOpenSourceContrib','q20CandPrevWorkExp','q20CandGithubPersProj2']]
```

Here we create a columns isNonTechnology to mark managers belong to Technology represented by 0 and managers from other industries represented by a 1

In [125]:
```
dfManagers['q3Gender'] = dfManagers['q3Gender'].apply(lambda x: 0 if x=='Male'
 else 1)
dfManagers['isNonTechnology'] = dfManagers['q10Industry'].apply(lambda x: 0 if
 x=='Technology' else 1 )
dfManagers.drop('q10Industry',axis=1,inplace=True)
```

In [126]:
```
Y = dfManagers.isNonTechnology
X = dfManagers.drop('isNonTechnology',axis=1)
```

In [127]:
```
import sklearn.tree
dt = sklearn.tree.DecisionTreeClassifier(max_depth = 3)
```

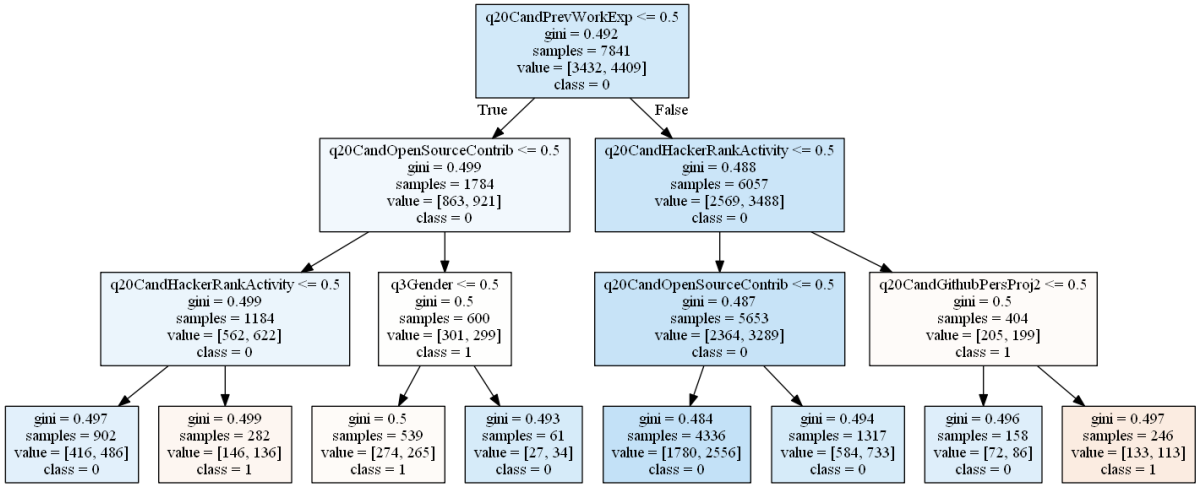
In [128]:
```
dt.fit(X,Y)
```

Out[128]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=3,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

Visualize the tree

In [129]:
```
# This code will visualize a decision tree dt,
# trained with the attributes in X and the class labels in Y
dt_feature_names = list(X.columns)
dt_target_names = np.array(Y.unique(),dtype=np.string_)
result = np.array([s.decode('UTF-8') for s in dt_target_names])
tree.export_graphviz(dt, out_file='tree.dot',
    feature_names=dt_feature_names, class_names=result,filled=True)
graph = pydotplus.graph_from_dot_file('tree.dot')
Image(graph.create_png())
```

Out[129]:



As we can see, on level 3 where Candidates github projects are valued the class changes to 1 which are for managers belonging to industries other than technology.