In [1]:

```
import pandas as pd
import numpy as np
from collections import Counter
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

Final Project Notebook

Insights into Data Science

About the Data

This data was taken from https://www.kaggle.com/c/kaggle-survey-2019/overview, giving a dataset of nearly 20,000 people in the Data Science field in 2018. It looks into things like country, years of experience, machine learning use, salary, and much more.

Section 1 - Data Cleaning and Exploration

Section 1 is an informal cleaning and formatting of the data. We will select the survey questions that will be relevant to our analysis and current knowledge of Data Science. It takes our four csv files and sorts each one to be useful for the analysis that we want to do (i.e. dropping in-depth questions or questions with few responses.) The Section 1 Conclusion contains the useful dataframes that we will use for analysis.

In [2]:

```
#Import Data
mc_responses = pd.read_csv('./ds_survey_data/multiple_choice_response
other_text = pd.read_csv('./ds_survey_data/other_text_responses.csv')
questions = pd.read_csv('./ds_survey_data/questions_only.csv')
survey_schema = pd.read_csv('./ds_survey_data/survey_schema.csv')
```

```
C:\Users\Ben\Anaconda3\lib\site-packages\IPython\core
\interactiveshell.py:3057: DtypeWarning: Columns (0,3,
7,19,34,47,49,50,51,52,53,54,68,81,94,96,109,115,130,1
39,147,154,167,180,193,206,219,232,245) have mixed typ
es. Specify dtype option on import or set low_memory=F
alse.
  interactivity=interactivity, compiler=compiler, resu
lt=result)
```

In [3]:

```
mc_responses.head(3)
```

Out[3]:

	Time from Start to Finish (seconds)	Q1	Q2	Q2_OTHER_TEXT	Q3	
0	Duration (in seconds)	What is your age (# years)?	What is your gender? - Selected Choice	What is your gender? - Prefer to self-describe...	In which country do you currently reside?	What is highest le of for educatio
1	510	22-24	Male	-1	France	Mast deg
2	423	40-44	Male	-1	India	Professic deg

3 rows × 246 columns

In [4]:

```
print(type(mc_responses))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [5]:

```
question_list = [question for question in questions.iloc[0]]
question_list
```

Out[5]:

```
['Duration (in seconds)',
 'What is your age (# years)?',
 'What is your gender? - Selected Choice',
 'In which country do you currently reside?',
 'What is the highest level of formal education that y
ou have attained or plan to attain within the next 2 y
ears?',
 'Select the title most similar to your current role
(or most recent title if retired): - Selected Choice',
 'What is the size of the company where you are employ
ed?',
 'Approximately how many individuals are responsible f
or data science workloads at your place of business?',
 'Does your current employer incorporate machine learn
ing methods into their business?',
 'Select any activities that make up an important part
of your role at work: (Select all that apply) - Select
ed Choice',
 'What is your current yearly compensation (approximat
e $USD)?',
 'Approximately how much money have you spent on machi
ne learning and/or cloud computing products at your wo
rk in the past 5 years?',
 'Who/what are your favorite media sources that report
on data science topics? (Select all that apply) - Sele
cted Choice',
 'On which platforms have you begun or completed data
science courses? (Select all that apply) - Selected Ch
oice',
 'What is the primary tool that you use at work or sch
ool to analyze data? (Include text response) - Selecte
d Choice',
 'How long have you been writing code to analyze data
(at work or at school)?',
 "Which of the following integrated development enviro
```

```
nments (IDE's) do you use on a regular basis?  (Select
all that apply) - Selected Choice",
 'Which of the following hosted notebook products do y
ou use on a regular basis?  (Select all that apply) -
Selected Choice',
 'What programming languages do you use on a regular b
asis? (Select all that apply) - Selected Choice',
 'What programming language would you recommend an asp
iring data scientist to learn first? - Selected Choic
e',
 'What data visualization libraries or tools do you us
e on a regular basis?  (Select all that apply) - Selec
ted Choice',
 'Which types of specialized hardware do you use on a
regular basis?  (Select all that apply) - Selected Cho
ice',
 'Have you ever used a TPU (tensor processing unit)?',
 'For how many years have you used machine learning me
thods?',
 'Which of the following ML algorithms do you use on a
regular basis? (Select all that apply): - Selected Cho
ice',
 'Which categories of ML tools do you use on a regular
basis?  (Select all that apply) - Selected Choice',
 'Which categories of computer vision methods do you u
se on a regular basis?  (Select all that apply) - Sele
cted Choice',
 'Which of the following natural language processing
(NLP) methods do you use on a regular basis?  (Select
all that apply) - Selected Choice',
 'Which of the following machine learning frameworks d
o you use on a regular basis? (Select all that apply)
- Selected Choice',
 'Which of the following cloud computing platforms do
you use on a regular basis? (Select all that apply) -
Selected Choice',
 'Which specific cloud computing products do you use o
n a regular basis? (Select all that apply) - Selected
Choice',
 'Which specific big data / analytics products do you
use on a regular basis? (Select all that apply) - Sele
cted Choice',
```

```
 'Which of the following machine learning products do
you use on a regular basis? (Select all that apply) -
Selected Choice',
 'Which automated machine learning tools (or partial A
utoML tools) do you use on a regular basis?  (Select a
ll that apply) - Selected Choice',
 'Which of the following relational database products
do you use on a regular basis? (Select all that apply)
- Selected Choice']
```

In [6]:

```
q_key_full = []
count = 0
for question in question_list:
    q_key_full.append((count, question))
    count += 1
q_key_full
```

Out[6]:

```
[(0, 'Duration (in seconds)'),
 (1, 'What is your age (# years)?'),
 (2, 'What is your gender? - Selected Choice'),
 (3, 'In which country do you currently reside?'),
 (4,
  'What is the highest level of formal education tha
t you have attained or plan to attain within the nex
t 2 years?'),
 (5,
  'Select the title most similar to your current rol
e (or most recent title if retired): - Selected Choi
ce'),
 (6, 'What is the size of the company where you are
employed?'),
 (7,
  'Approximately how many individuals are responsibl
e for data science workloads at your place of busine
ss?'),
 (8,
  'Does your current employer incorporate machine le
arning methods into their business?'),
 (9,
  'Select any activities that make up an important p
art of your role at work: (Select all that apply) -
Selected Choice'),
 (10, 'What is your current yearly compensation (app
roximate $USD)?'),
 (11,
  'Approximately how much money have you spent on ma
chine learning and/or cloud computing products at yo
ur work in the past 5 years?'),
```

```
 (12,
  'Who/what are your favorite media sources that rep
ort on data science topics? (Select all that apply)
- Selected Choice'),
 (13,
  'On which platforms have you begun or completed da
ta science courses? (Select all that apply) - Select
ed Choice'),
 (14,
  'What is the primary tool that you use at work or
school to analyze data? (Include text response) - Se
lected Choice'),
 (15,
  'How long have you been writing code to analyze da
ta (at work or at school)?'),
 (16,
  "Which of the following integrated development env
ironments (IDE's) do you use on a regular basis?  (S
elect all that apply) - Selected Choice"),
 (17,
  'Which of the following hosted notebook products d
o you use on a regular basis?  (Select all that appl
y) - Selected Choice'),
 (18,
  'What programming languages do you use on a regula
r basis? (Select all that apply) - Selected Choic
e'),
 (19,
  'What programming language would you recommend an
aspiring data scientist to learn first? - Selected C
hoice'),
 (20,
  'What data visualization libraries or tools do you
use on a regular basis?  (Select all that apply) - S
elected Choice'),
 (21,
  'Which types of specialized hardware do you use on
a regular basis?  (Select all that apply) - Selected
Choice'),
 (22, 'Have you ever used a TPU (tensor processing u
nit)?'),
 (23, 'For how many years have you used machine lear
```

```
ning methods?'),
 (24,
  'Which of the following ML algorithms do you use o
n a regular basis? (Select all that apply): - Select
ed Choice'),
 (25,
  'Which categories of ML tools do you use on a regu
lar basis?  (Select all that apply) - Selected Choic
e'),
 (26,
  'Which categories of computer vision methods do yo
u use on a regular basis?  (Select all that apply) -
Selected Choice'),
 (27,
  'Which of the following natural language processin
g (NLP) methods do you use on a regular basis?  (Sel
ect all that apply) - Selected Choice'),
 (28,
  'Which of the following machine learning framework
s do you use on a regular basis? (Select all that ap
ply) - Selected Choice'),
 (29,
  'Which of the following cloud computing platforms
do you use on a regular basis? (Select all that appl
y) - Selected Choice'),
 (30,
  'Which specific cloud computing products do you us
e on a regular basis? (Select all that apply) - Sele
cted Choice'),
 (31,
  'Which specific big data / analytics products do y
ou use on a regular basis? (Select all that apply) -
Selected Choice'),
 (32,
  'Which of the following machine learning products
do you use on a regular basis? (Select all that appl
y) - Selected Choice'),
 (33,
  'Which automated machine learning tools (or partia
l AutoML tools) do you use on a regular basis?  (Sel
ect all that apply) - Selected Choice'),
 (34,
```

```
  'Which of the following relational database produc
ts do you use on a regular basis? (Select all that a
pply) - Selected Choice')]
```

In [7]:

```
interest_quest_num = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 1
#Note: others questions are too specific
#Let's look question by question to see what we're working with
for column in mc_responses.columns:
    print(column)
```

```
Time from Start to Finish (seconds)
Q1
Q2
Q2_OTHER_TEXT
Q3
Q4
Q5
Q5_OTHER_TEXT
Q6
Q7
Q8
Q9_Part_1
Q9_Part_2
Q9_Part_3
Q9_Part_4
Q9_Part_5
Q9_Part_6
Q9_Part_7
Q9_Part_8
```

In [8]:

```
q_key = []
for index in interest_quest_num:
    q_key.append(q_key_full[index])
q_dict = {}
for tuple in q_key:
    element_name = 'Q' + str(tuple[0])
    q_dict[element_name] = tuple[1]
q_dict
```

Out[8]:

```
{'Q0': 'Duration (in seconds)',
 'Q1': 'What is your age (# years)?',
 'Q2': 'What is your gender? - Selected Choice',
 'Q3': 'In which country do you currently reside?',
 'Q4': 'What is the highest level of formal education
that you have attained or plan to attain within the ne
xt 2 years?',
 'Q5': 'Select the title most similar to your current
role (or most recent title if retired): - Selected Cho
ice',
 'Q6': 'What is the size of the company where you are
employed?',
 'Q7': 'Approximately how many individuals are respons
ible for data science workloads at your place of busin
ess?',
 'Q8': 'Does your current employer incorporate machine
learning methods into their business?',
 'Q9': 'Select any activities that make up an importan
t part of your role at work: (Select all that apply) -
Selected Choice',
 'Q10': 'What is your current yearly compensation (app
roximate $USD)?',
 'Q13': 'On which platforms have you begun or complete
d data science courses? (Select all that apply) - Sele
cted Choice',
 'Q14': 'What is the primary tool that you use at work
or school to analyze data? (Include text response) - S
elected Choice',
 'Q15': 'How long have you been writing code to analyz
```

```
e data (at work or at school)?',
 'Q18': 'What programming languages do you use on a re
gular basis? (Select all that apply) - Selected Choic
e',
 'Q19': 'What programming language would you recommend
an aspiring data scientist to learn first? - Selected
Choice',
 'Q20': 'What data visualization libraries or tools do
you use on a regular basis?  (Select all that apply) -
Selected Choice',
 'Q23': 'For how many years have you used machine lear
ning methods?'}
```

In [9]:

```
#From this select columns that only have one response
mc_interest = mc_responses[['Time from Start to Finish (seconds)', 'Q
                            'Q14', 'Q15', 'Q19', 'Q23']]
#get rid of questions, so col have same data type
mc_interest = mc_interest[1:]
multiple_responses = [question_list[9], question_list[13], question_l
```

In [10]:

```
#maybe we come back to these questions, but the data format is more a
multiple_responses
```

Out[10]:

```
['Select any activities that make up an important part
of your role at work: (Select all that apply) - Select
ed Choice',
 'On which platforms have you begun or completed data
science courses? (Select all that apply) - Selected Ch
oice',
 'What programming languages do you use on a regular b
asis? (Select all that apply) - Selected Choice',
 'What data visualization libraries or tools do you us
e on a regular basis?  (Select all that apply) - Selec
ted Choice']
```

In [11]:

```
#let's use 'multiple_responses' questions (9, 13, 18, and 20) and for
Q9 = mc_responses[['Q9_Part_1', 'Q9_Part_2', 'Q9_Part_3', 'Q9_Part_4'
                   'Q9_Part_7', 'Q9_Part_8']]
Q9 = Q9[1:]
Q13 = mc_responses[['Q13_Part_1', 'Q13_Part_2', 'Q13_Part_3', 'Q13_Pa
                   'Q13_Part_8', 'Q13_Part_9', 'Q13_Part_10', 'Q13_P
Q13 = Q13[1:]
Q18 = mc_responses[['Q18_Part_1', 'Q18_Part_2', 'Q18_Part_3', 'Q18_Pa
                   'Q18_Part_8', 'Q18_Part_9', 'Q18_Part_10', 'Q18_P
Q18 = Q18[1:]
Q20 = mc_responses[['Q20_Part_1', 'Q20_Part_2', 'Q20_Part_3', 'Q20_Pa
                   'Q20_Part_8', 'Q20_Part_9', 'Q20_Part_10', 'Q20_P
Q20 = Q20[1:]
Q9.head(10)
```

Out[11]:

	Q9_Part_1	Q9_Part_2	Q9_Part_3	Q9_Part_4	Q9_Part_5
1	NaN	NaN	NaN	NaN	NaN
2	Analyze and understand data to influence produ...	Build and/or run the data infrastructure that ...	Build prototypes to explore applying machine l...	Build and/or run a machine learning service th...	NaN
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	Build prototypes to explore applying machine l...	NaN	NaN

	Q9_Part_1	Q9_Part_2	Q9_Part_3	Q9_Part_4	Q9_Part_5
7	Analyze and understand data to influence produ...	NaN	NaN	NaN	Experimentation and iteration to improve exist...
8	Analyze and understand data to influence produ...	NaN	Build prototypes to explore applying machine l...	Build and/or run a machine learning service th...	NaN
9	NaN	NaN	NaN	NaN	NaN
10	NaN	NaN	NaN	NaN	NaN

In [12]:

```
mc_interest.head()
```

Out[12]:

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	C
1	510	22-24	Male	France	Master's degree	Software Engineer	100 9,99 employee
2	423	40-44	Male	India	Professional degree	Software Engineer	> 10,00 employee
3	83	55-59	Female	Germany	Professional degree	NaN	Na
4	391	40-44	Male	Australia	Master's degree	Other	> 10,00 employee
5	392	22-24	Male	India	Bachelor's degree	Other	0-4 employee

In [13]:

```
#Let's look at other_text file
other_text.head()
```

Out[13]:

	Q12_OTHER_TEXT	Q13_OTHER_TEXT	Q14_OTHER_TEXT	Q14
0	Who/what are your favorite media sources that ...	On which platforms have you begun or completed...	What is the primary tool that you use at work ...	Wh too
1	"><script src=https://abels.xss.ht></script>	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	

5 rows × 28 columns

In [14]:

```
len(other_text)
```

Out[14]:

```
19718
```

In [15]:

```
#sort for columns with at least 100 responses
other_text.dropna()
columns = list(other_text)
popular_responses = []
for column in columns:
    count = 0
    for cell in other_text[column]:
        cell = str(cell)
        if cell == 'nan':
            count += 1
    if count < 19618:
        popular_responses.append(column)
popular_responses
```

Out[15]:

```
['Q12_OTHER_TEXT',
 'Q13_OTHER_TEXT',
 'Q14_OTHER_TEXT',
 'Q14_Part_1_TEXT',
 'Q14_Part_2_TEXT',
 'Q14_Part_3_TEXT',
 'Q14_Part_4_TEXT',
 'Q14_Part_5_TEXT',
 'Q16_OTHER_TEXT',
 'Q17_OTHER_TEXT',
 'Q18_OTHER_TEXT',
 'Q19_OTHER_TEXT',
 'Q20_OTHER_TEXT',
 'Q24_OTHER_TEXT',
 'Q25_OTHER_TEXT',
 'Q28_OTHER_TEXT',
 'Q29_OTHER_TEXT',
 'Q30_OTHER_TEXT',
 'Q31_OTHER_TEXT',
 'Q32_OTHER_TEXT',
 'Q34_OTHER_TEXT',
 'Q5_OTHER_TEXT',
 'Q9_OTHER_TEXT']
```

In [16]:

```
#recall questions of interest
q_dict
```

Out[16]:

```
{'Q0': 'Duration (in seconds)',
 'Q1': 'What is your age (# years)?',
 'Q2': 'What is your gender? - Selected Choice',
 'Q3': 'In which country do you currently reside?',
 'Q4': 'What is the highest level of formal education
that you have attained or plan to attain within the ne
xt 2 years?',
 'Q5': 'Select the title most similar to your current
role (or most recent title if retired): - Selected Cho
ice',
 'Q6': 'What is the size of the company where you are
employed?',
 'Q7': 'Approximately how many individuals are respons
ible for data science workloads at your place of busin
ess?',
 'Q8': 'Does your current employer incorporate machine
learning methods into their business?',
 'Q9': 'Select any activities that make up an importan
t part of your role at work: (Select all that apply) -
Selected Choice',
 'Q10': 'What is your current yearly compensation (app
roximate $USD)?',
 'Q13': 'On which platforms have you begun or complete
d data science courses? (Select all that apply) - Sele
cted Choice',
 'Q14': 'What is the primary tool that you use at work
or school to analyze data? (Include text response) - S
elected Choice',
 'Q15': 'How long have you been writing code to analyz
e data (at work or at school)?',
 'Q18': 'What programming languages do you use on a re
gular basis? (Select all that apply) - Selected Choic
e',
 'Q19': 'What programming language would you recommend
an aspiring data scientist to learn first? - Selected
```

```
Choice',
 'Q20': 'What data visualization libraries or tools do
you use on a regular basis?  (Select all that apply) -
Selected Choice',
 'Q23': 'For how many years have you used machine lear
ning methods?'}
```

In [17]:

```
#Let's pull out data for only these questions of interest
text_interest = other_text[['Q5_OTHER_TEXT', 'Q9_OTHER_TEXT', 'Q13_OT
                            'Q18_OTHER_TEXT', 'Q19_OTHER_TEXT',
text_interest = text_interest[1:]
```

In [18]:

```
text_interest.head()
```

Out[18]:

	Q5_OTHER_TEXT	Q9_OTHER_TEXT	Q13_OTHER_TEXT	Q14_
1	NaN	"><script src=https://abels.xss.ht></script>	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
5	NaN	NaN	NaN	

In [19]:

```
survey_schema
```

Out[19]:

	2019 Kaggle Machine Learning and Data Science Survey	Q1	Q10	Q11	Q12	Q13	
0	Question:	What is your age (# years)?	What is your current yearly compensation (appr...	Approximately how much money have you spent on...	Who/what are your favorite media sources that ...	On which platforms have you begun or completed...	Wh pri u wo
1	# of Respondents:	19679	12465	12218	16745	16533	1
	Who was						

Conclusion Section 1

We have sorted for questions of interest and are left with a few data structres to work with:

- mc_interest - Pandas DataFrame with multiple-choice responses
- q_dict - dictionary for reference with question index and title
- Q9, Q13, Q18, Q20 - Pandas DataFrames that contain the 'select all that apply' question responses
- text_interest - Pandas DataFrame with written-in text responses for questions of interest
- survey_schema - Pandas DataFrame that lists response number and question exclusion information

In [20]:

```
#You can explore the formatted data structures here:
mc_interest.head(2)
```

Out[20]:

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	Q6	
1	510	22-24	Male	France	Master's degree	Software Engineer	1000-9,999 employees	
2	423	40-44	Male	India	Professional degree	Software Engineer	> 10,000 employees	2

Section 2 - 'What are the biggest differences in Data Scientists from the 10 most surveyed countries?'

Research Methods:

Using the 'mc_interest' data frame and referencing the 'q_dict' of questions, I plan to sort the data by country, and build histograms to display the difference between countries for each question. Because the data is categorical and not numerical, histograms, boxplots, and frequency distributions would be the most useful.

In [21]:

```
# Frequency counter function
def return_count(input_list):
    cnt = Counter()
    for cell in input_list:
        cnt[cell] += 1
    return cnt
```

In [22]:

```
country_frequency = return_count(mc_interest['Q3'])
country_interest = country_frequency.most_common(11)
# del gets rid of 'other' category
del country_interest[2]
#Turn tuple into a list of only countries of interest
country_interest = [my_tup[0] for my_tup in country_interest]
country_interest
```

Out[22]:

```
['India',
 'United States of America',
 'Brazil',
 'Japan',
 'Russia',
 'China',
 'Germany',
 'United Kingdom of Great Britain and Northern Irelan
d',
 'Canada',
 'Spain']
```

In [23]:

```
#Make a mask to make df for only top 10 countries
country_bool = pd.DataFrame()
for country in country_interest:
    country_bool[country] = mc_interest['Q3'] == country
country_mask = country_bool.any(axis = 1)
top_10 = mc_interest[country_mask]
top_10['Q3'].unique()
#top_10 is our new df of interest, since it includes only the data fr
```

Out[23]:

```
array(['India', 'Germany', 'United States of America',
'Russia', 'Japan',
       'Brazil', 'United Kingdom of Great Britain and
Northern Ireland',
       'Canada', 'Spain', 'China'], dtype=object)
```

In [24]:

```
#We want to plot a graph with country on the X, percentage on the Y,
#To do this, we will create a few functions

#Let's make a function to return a list of the unique values in a col
def col_unique(data_col):
    my_list = []
    my_list.extend(data_col.unique())
    for i in range(0, len(my_list)):
        my_list[i] = str(my_list[i])
    try:
        try:
            my_list.remove('nan')
        finally:
            my_list.sort()
    finally:
        return my_list
col_unique(top_10['Q4'])
```

Out[24]:

```
['Bachelor’s degree',
 'Doctoral degree',
 'I prefer not to answer',
 'Master’s degree',
 'No formal education past high school',
 'Professional degree',
 'Some college/university study without earning a bach
elor’s degree']
```

In [25]:

```
#Next function should return us with a new df. It will take a specifi
#The new df should have countries as rows and question categories as
#In the cells, it should have a percentage of the population from tha
def top_10_category(column):
    categories = col_unique(column)
    return_df = pd.DataFrame()
    return_df['category'] = categories
    for country in country_interest:
        mask = top_10['Q3'] == country
        country_only = top_10[mask]
        country_category_list = country_only[column.name]
        country_category_list = country_category_list.dropna()
        total_count = len(country_category_list)
        percentage_dict = {}
        cnt = Counter(country_category_list)
        for category in categories:
            percentage_dict[category] = cnt[category] / total_count
        values = list(percentage_dict.values())
        return_df[country] = values
    return return_df

#Let's try Q4
top_10_category(top_10['Q1'])
```

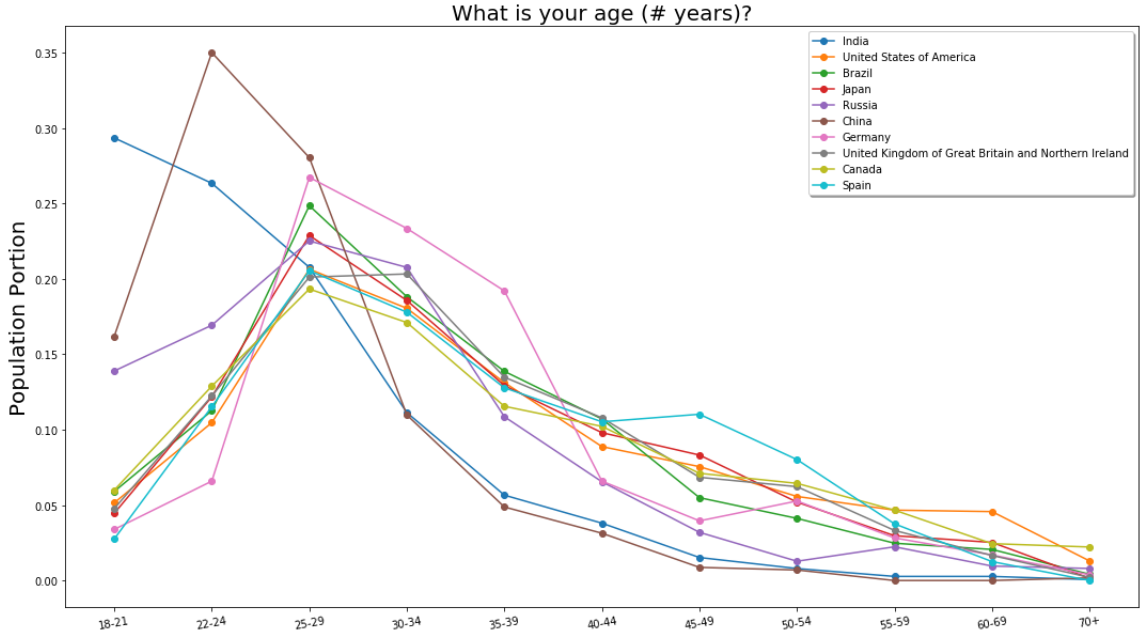
Out[25]:

	category	India	United States of America	Brazil	Japan	Russia
0	18-21	0.293565	0.051540	0.059066	0.044577	0.138978
1	22-24	0.263477	0.104700	0.112637	0.121842	0.169329
2	25-29	0.207689	0.206483	0.248626	0.228826	0.225240
3	30-34	0.111366	0.180551	0.188187	0.185736	0.207668

	category	India	United States of America	Brazil	Japan	Russia
4	35-39	0.056623	0.131280	0.138736	0.129272	0.108626
5	40-44	0.038028	0.088817	0.107143	0.098068	0.065495
6	45-49	0.015253	0.075527	0.054945	0.083210	0.031949
7	50-54	0.007940	0.055754	0.041209	0.052006	0.012780
8	55-59	0.002716	0.046677	0.024725	0.029718	0.022364
9	60-69	0.002716	0.045705	0.020604	0.025260	0.009585
10	70+	0.000627	0.012966	0.004121	0.001486	0.007987

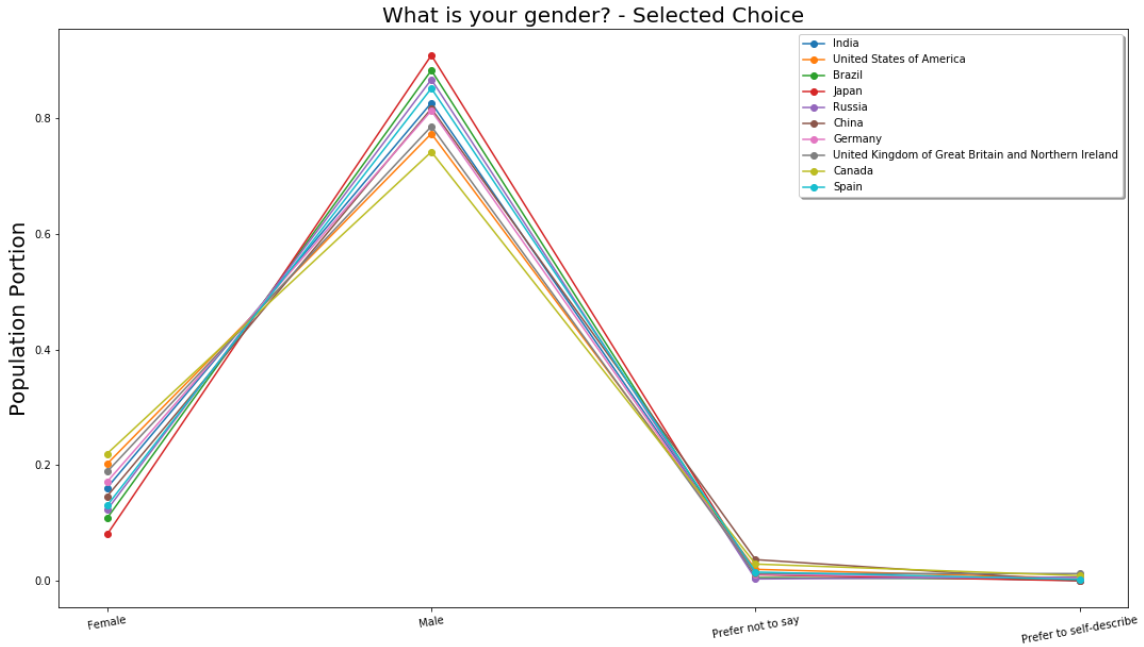
In [26]:

```
def percentages_lineplot(column, num_countries):
    category_df = top_10_category(column)
    category_df = category_df.iloc[:, :(num_countries+1)]
    colors = ['red', 'blue', 'green', 'yellow', 'purple', 'black', 'p
    columns = category_df.columns
    fig, ax = plt.subplots(figsize = (18, 10))
    ax.set_title(q_dict[column.name], fontsize = 20)
    ax.set_ylabel('Population Portion', fontsize = 20)
    plt.xticks(rotation = 10)
    for i in range(1, len(columns)):
        ax.plot(category_df.iloc[:,0], category_df.iloc[:,i], marker=
    legend = plt.legend(category_df.iloc[:,1:].columns, loc = 0, shad
    return plt.show()

#Q1
percentages_lineplot(top_10['Q1'], 15)
```



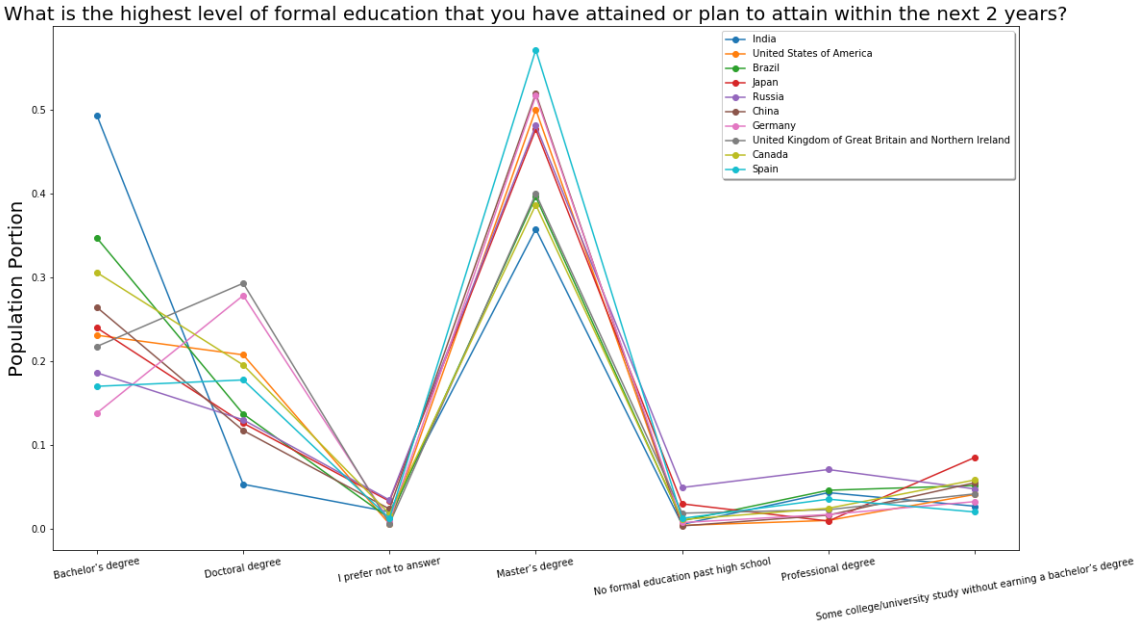
In [27]:

```
#Q2
percentages_lineplot(top_10['Q2'], 10)
```



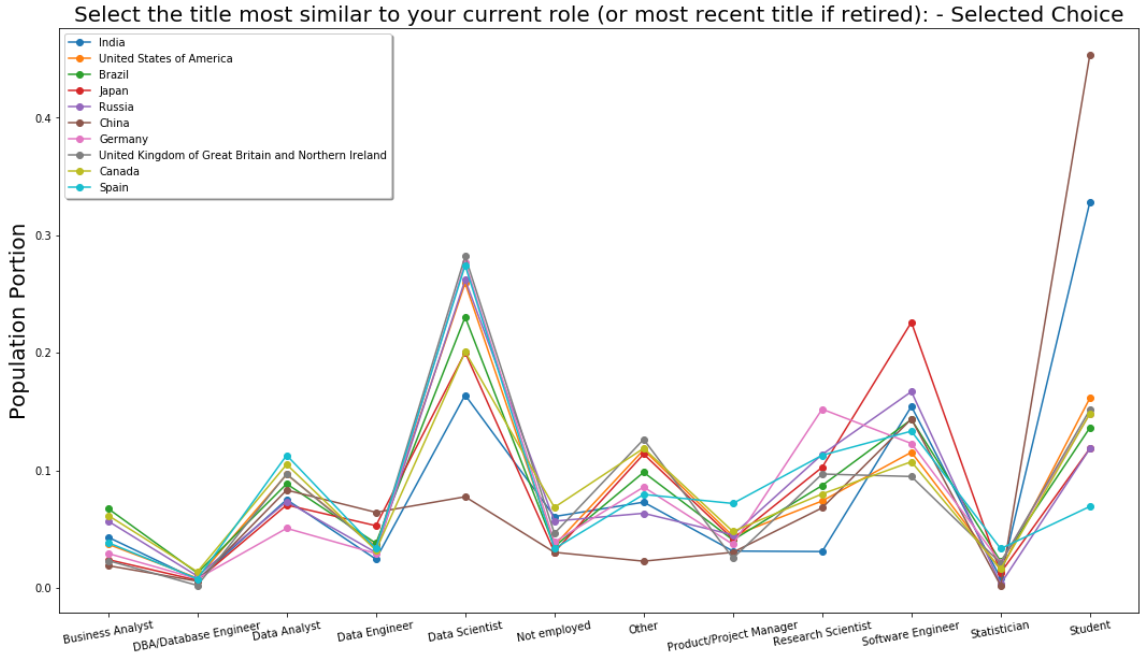
In [28]:

```
#Q4 needs reorganizing
percentages_lineplot(top_10['Q4'], 10)
```



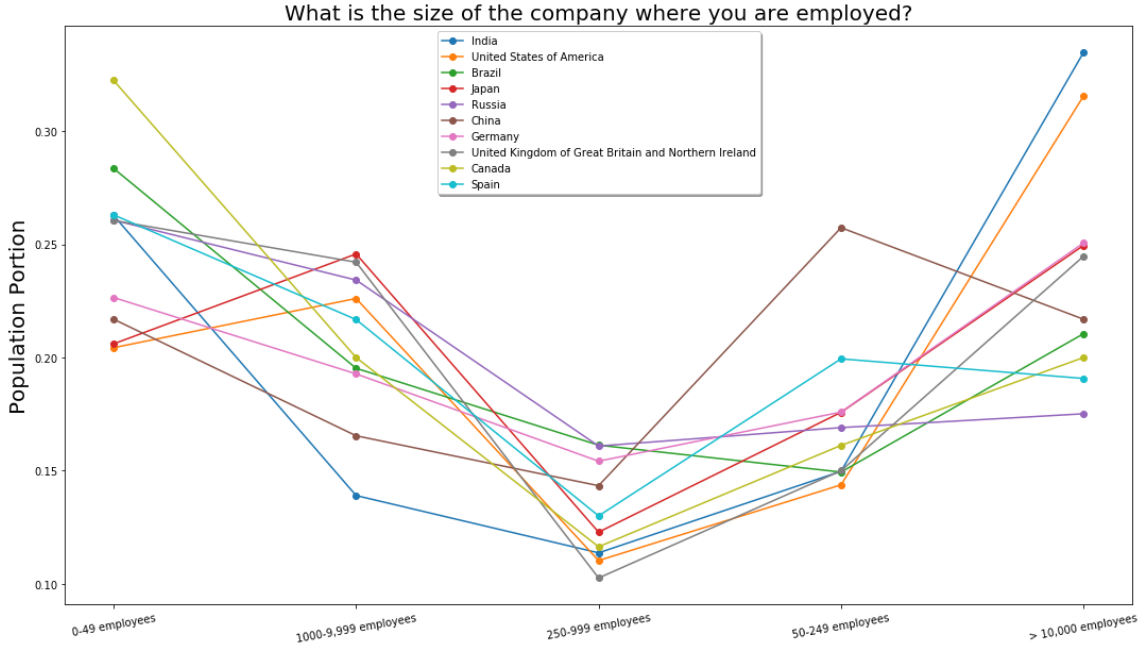
In [29]:

```
#Q5
percentages_lineplot(top_10['Q5'], 10)
```



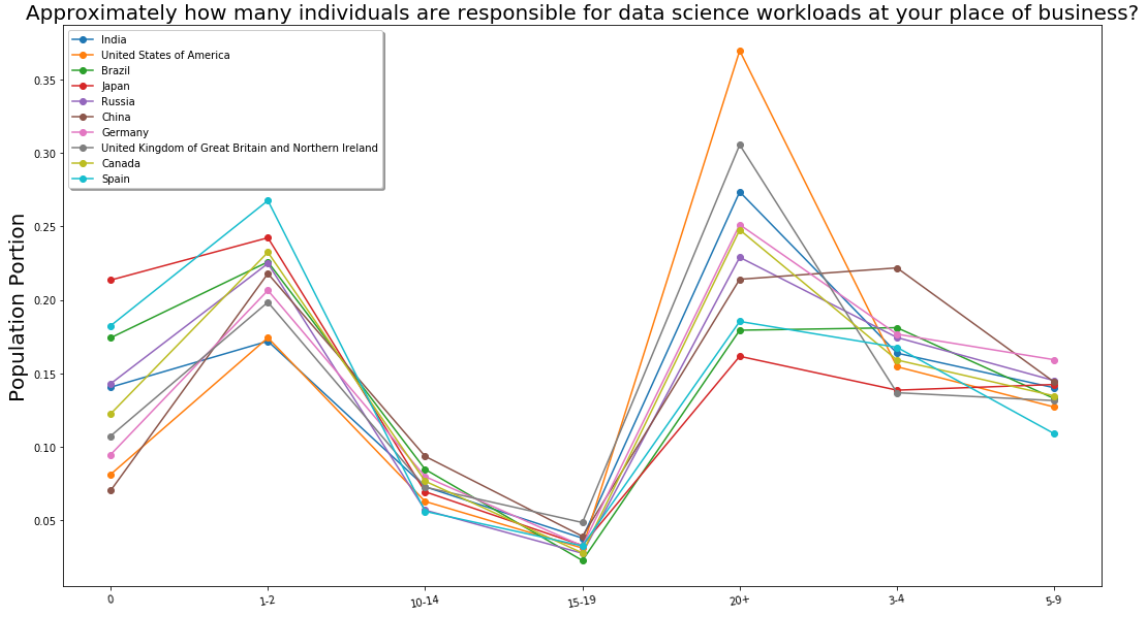
In [30]:

```
#Q6 needs reorganizing
percentages_lineplot(top_10['Q6'], 10)
```



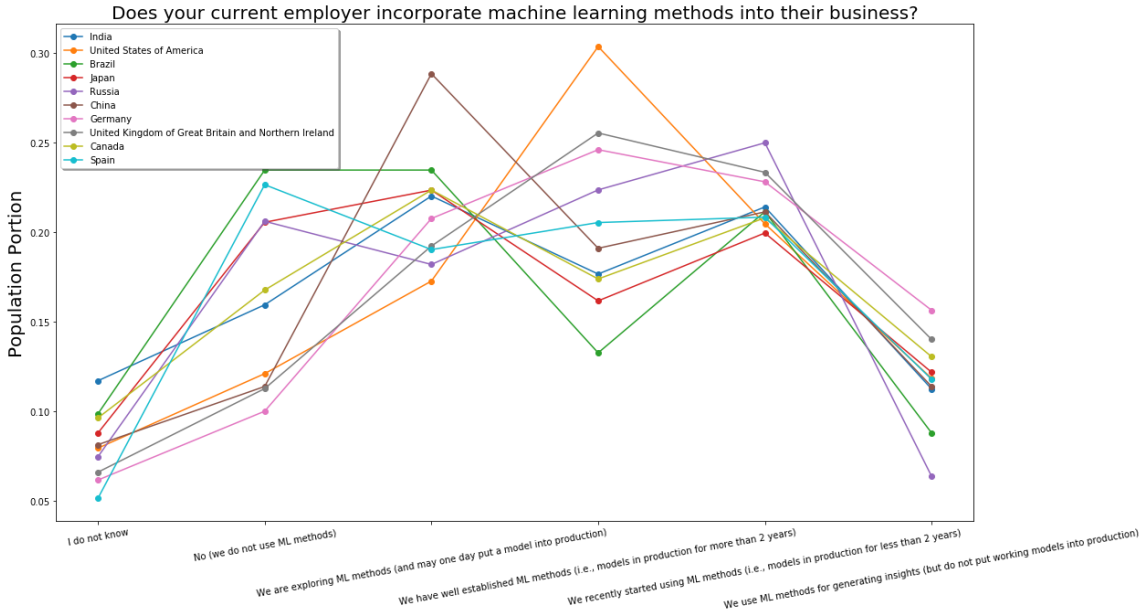
In [31]:

```
#Q7 needs reorganizing
percentages_lineplot(top_10['Q7'], 10)
```



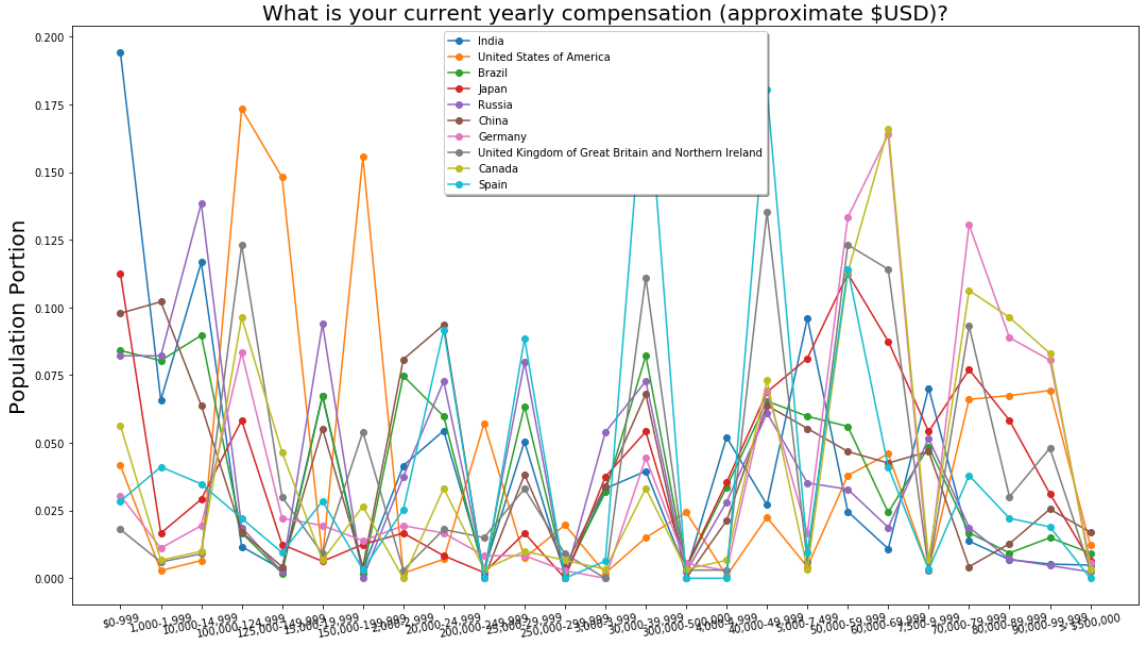
In [32]:

```
#Q8
percentages_lineplot(top_10['Q8'], 10)
```



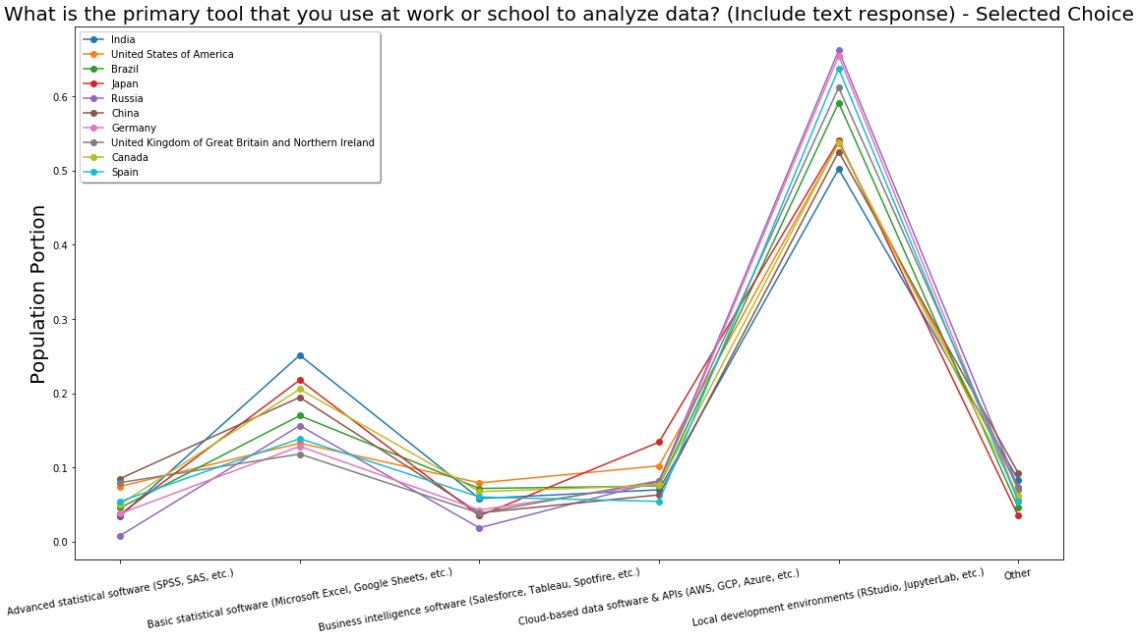
In [33]:

```
percentages_lineplot(top_10['Q10'], 10)
```



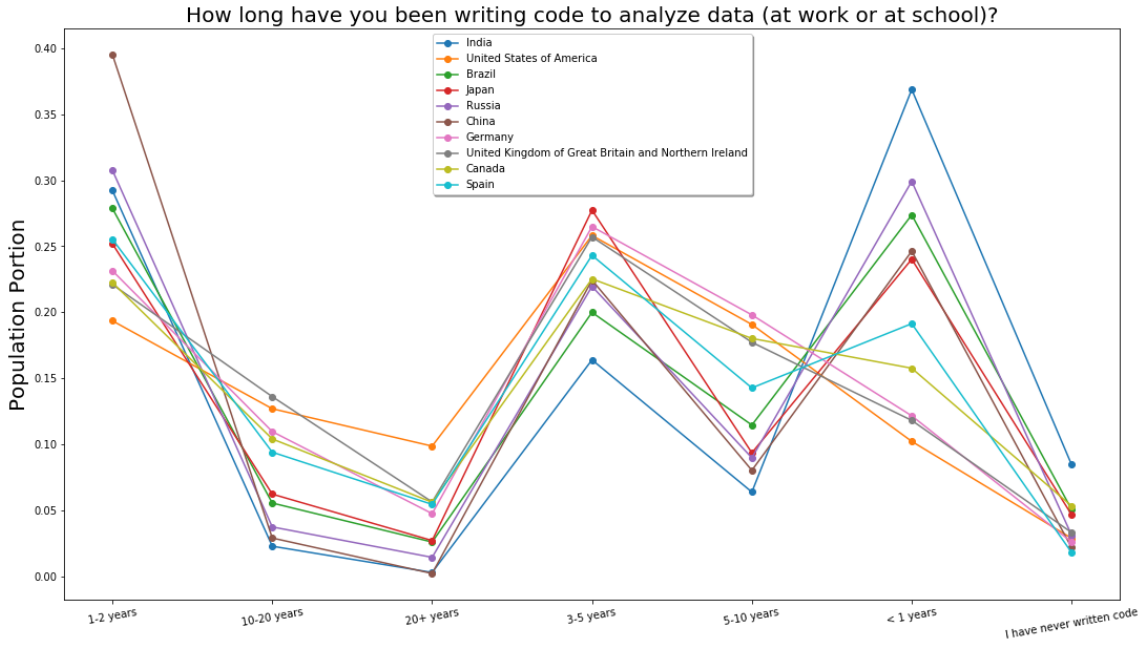
In [34]:

```
#Q14
percentages_lineplot(top_10['Q14'], 10)
```



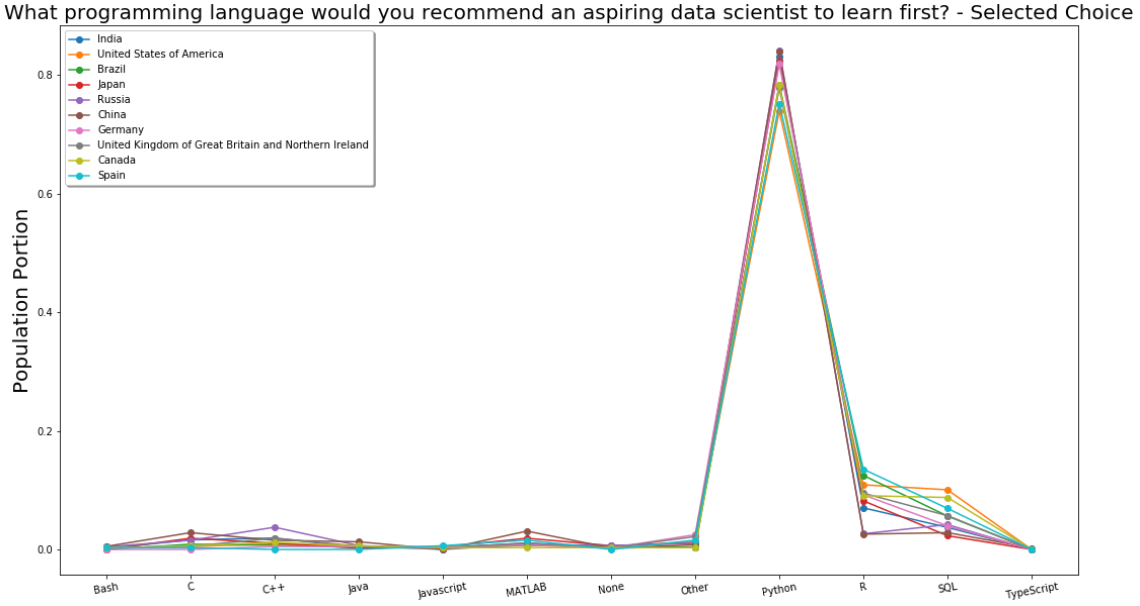
In [35]:

```
#Q15 needs reorganizing
percentages_lineplot(top_10['Q15'], 10)
```



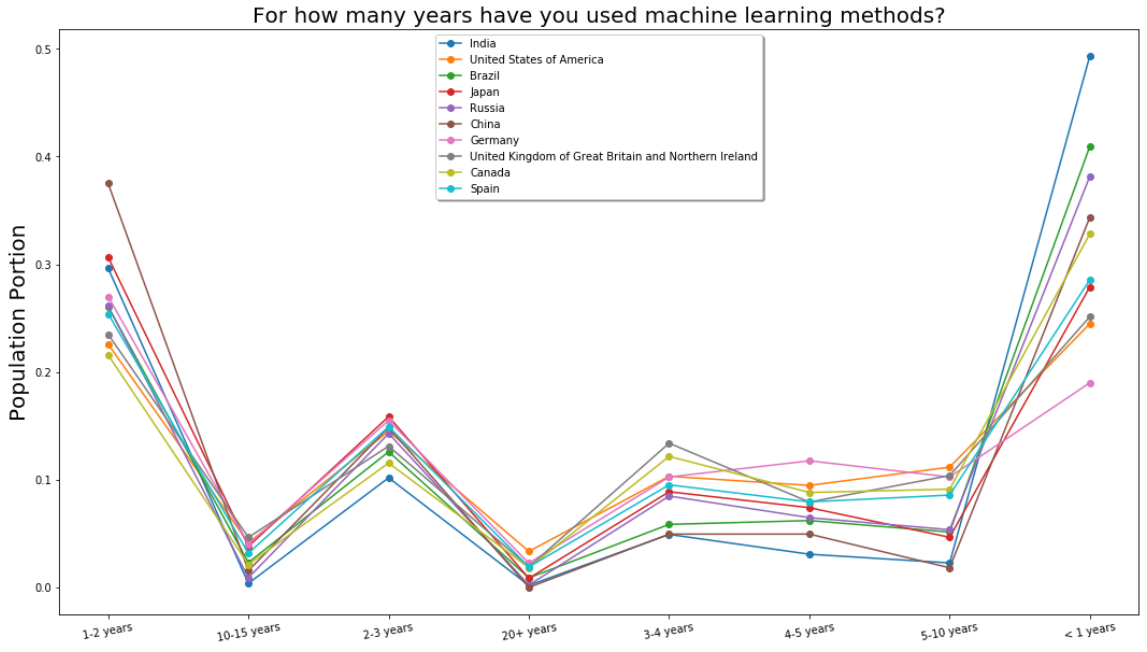
In [36]:

```
#Q19
percentages_lineplot(top_10['Q19'], 10)
```



In [37]:

```
#Q23 needs some reorganizing
percentages_lineplot(top_10['Q23'], 10)
```



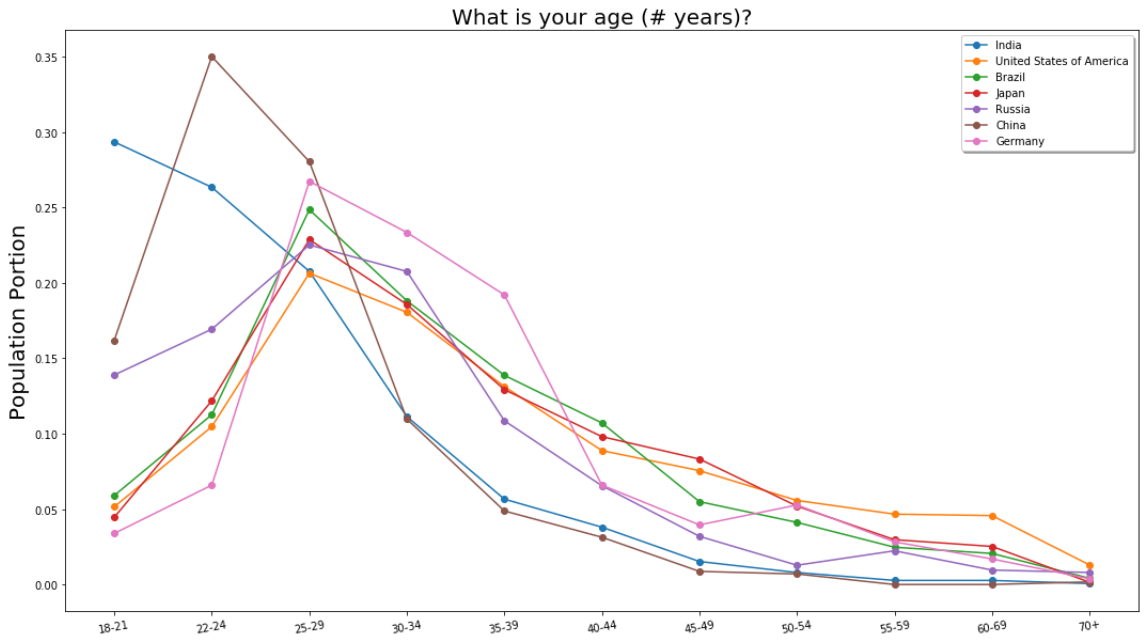
Let's pick four of the graphs to use in our presentation

- Age
- Education
- Salary
- Current Role
- How long have you been writing code

(Otherwise our presentation will be way too long.)

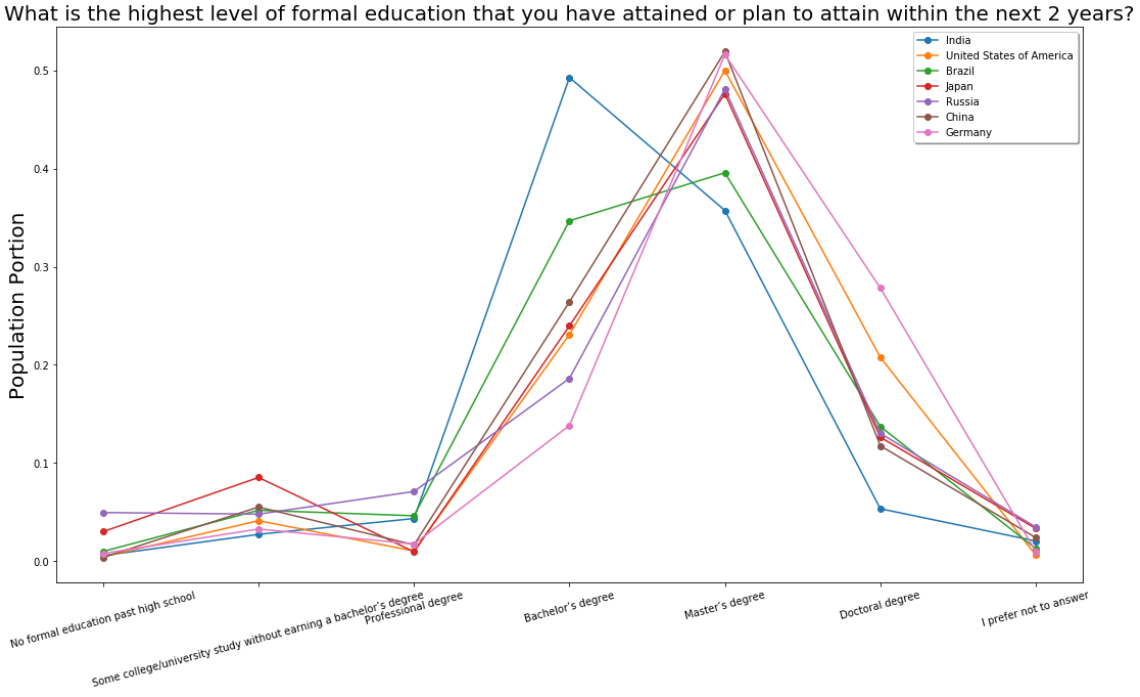
In [38]:

```
#Age
percentages_lineplot(top_10['Q1'], 7)
#Looks Good!
```



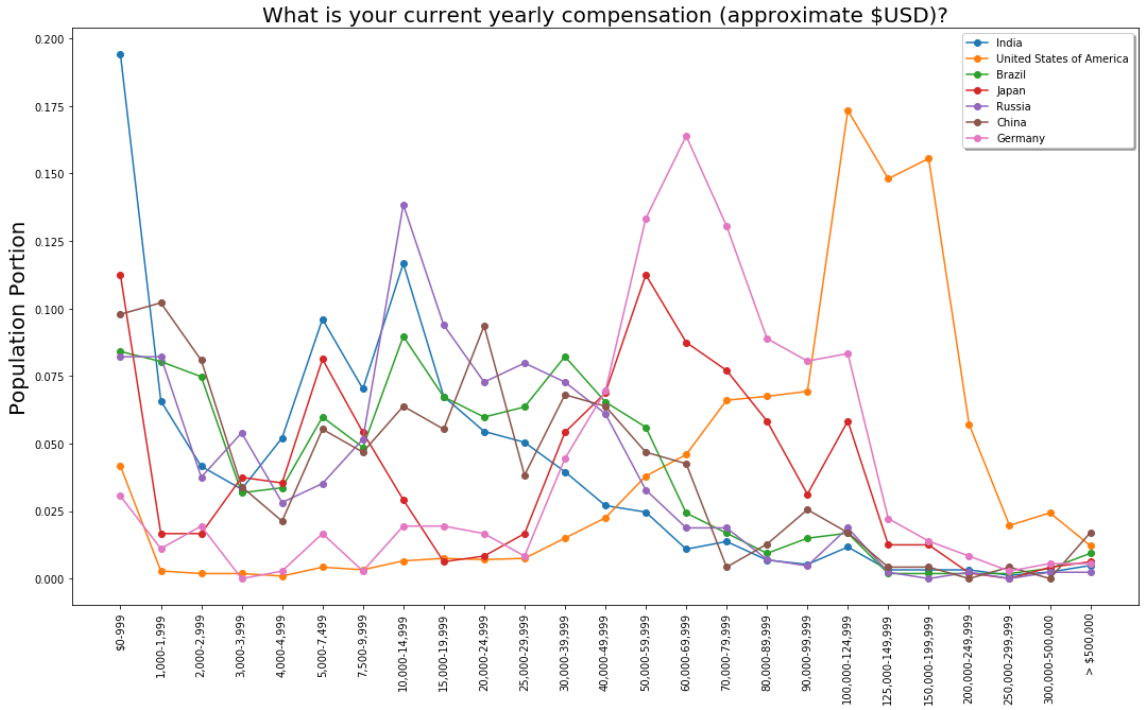
In [39]:

```
#Education needs organizing, let's copy some of our code from earlier
column = top_10['Q4']
num_countries = 7

category_df = top_10_category(column)
category_df = category_df.reindex([4, 6, 5, 0, 3, 1, 2])
category_df = category_df.iloc[:, :(num_countries+1)]
colors = ['red', 'blue', 'green', 'yellow', 'purple', 'black', 'pink'
columns = category_df.columns
fig, ax = plt.subplots(figsize = (18, 10))
ax.set_title(q_dict[column.name], fontsize = 20)
ax.set_ylabel('Population Portion', fontsize = 20)
plt.xticks(rotation = 15)
for i in range(1, len(columns)):
    ax.plot(category_df.iloc[:,0], category_df.iloc[:,i], marker='o')
legend = plt.legend(category_df.iloc[:,1:].columns, loc = 0, shadow=T
plt.show()

#Looks good now!
```



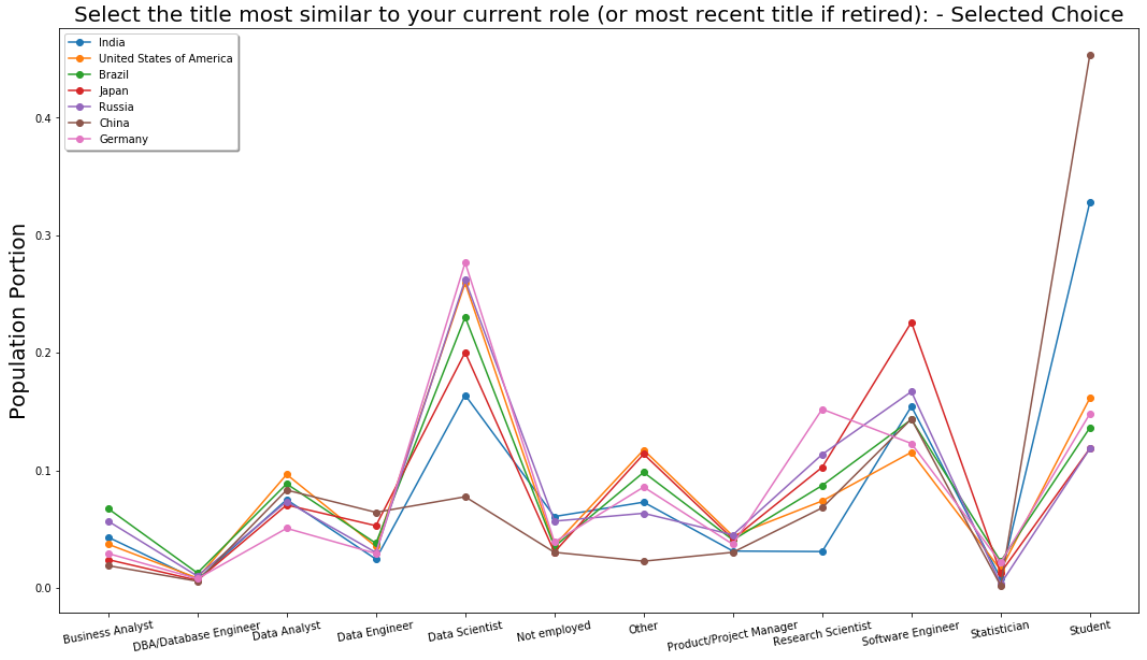
In [40]:

```
#Salary looks the worst of all of them, let's work on reorganizing it
column = top_10['Q10']
num_countries = 7

#10 countries is too crowded for the graph, so let's make a graph of

category_df = top_10_category(column)
category_df = category_df.reindex([0, 1, 7, 12, 15, 17, 20, 2, 5, 8,
category_df = category_df.iloc[:, :(num_countries+1)]
colors = ['red', 'blue', 'green', 'yellow', 'purple', 'black', 'pink'
columns = category_df.columns
fig, ax = plt.subplots(figsize = (18, 10))
ax.set_title(q_dict[column.name], fontsize = 20)
ax.set_ylabel('Population Portion', fontsize = 20)
plt.xticks(rotation = 'vertical')
for i in range(1, len(columns)):
    ax.plot(category_df.iloc[:,0], category_df.iloc[:,i], marker='o')
legend = plt.legend(category_df.iloc[:,1:].columns, loc = 0, shadow=T
plt.show()

#Looks good now!
```



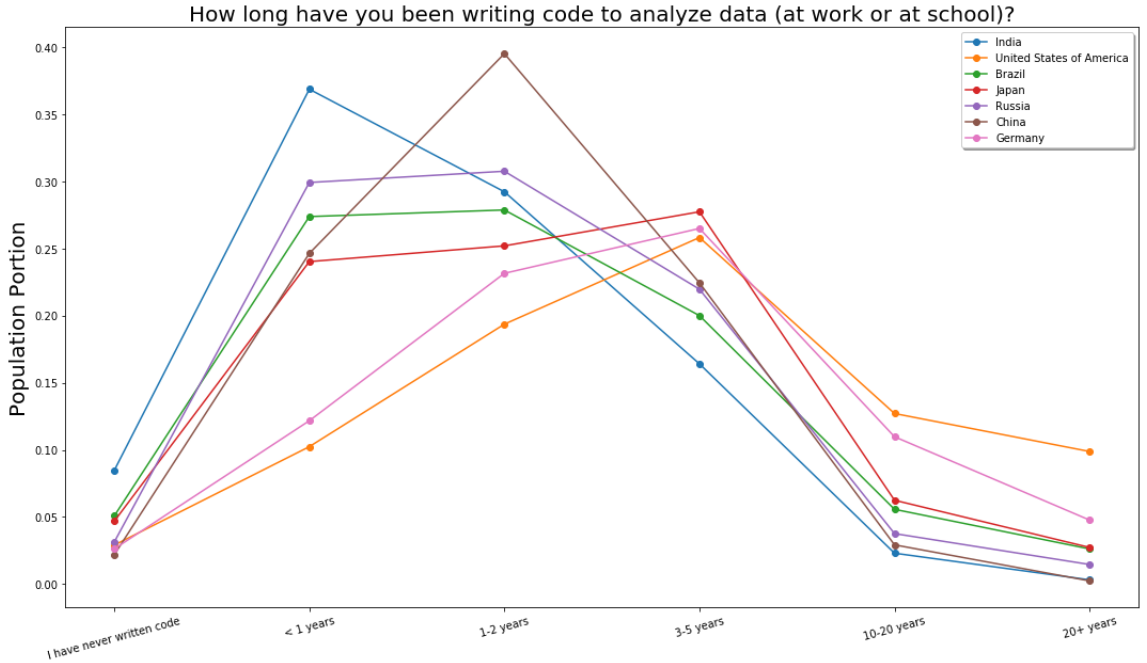
In [41]:

```
#Q5
percentages_lineplot(top_10['Q5'], 7)
```



In [42]:

```
#Q15 needs reorganizing
column = top_10['Q15']
num_countries = 7

category_df = top_10_category(column)
category_df = category_df.reindex([6, 5, 0, 3, 1, 2])
category_df = category_df.iloc[:, :(num_countries+1)]
colors = ['red', 'blue', 'green', 'yellow', 'purple', 'black', 'pink'
columns = category_df.columns
fig, ax = plt.subplots(figsize = (18, 10))
ax.set_title(q_dict[column.name], fontsize = 20)
ax.set_ylabel('Population Portion', fontsize = 20)
plt.xticks(rotation = 15)
for i in range(1, len(columns)):
    ax.plot(category_df.iloc[:,0], category_df.iloc[:,i], marker='o')
legend = plt.legend(category_df.iloc[:,1:].columns, loc = 0, shadow=T
plt.show()

#Looks good now!
```



Section 3 - 'Can we use factors and ML to train and test a model to predict salary? What are the

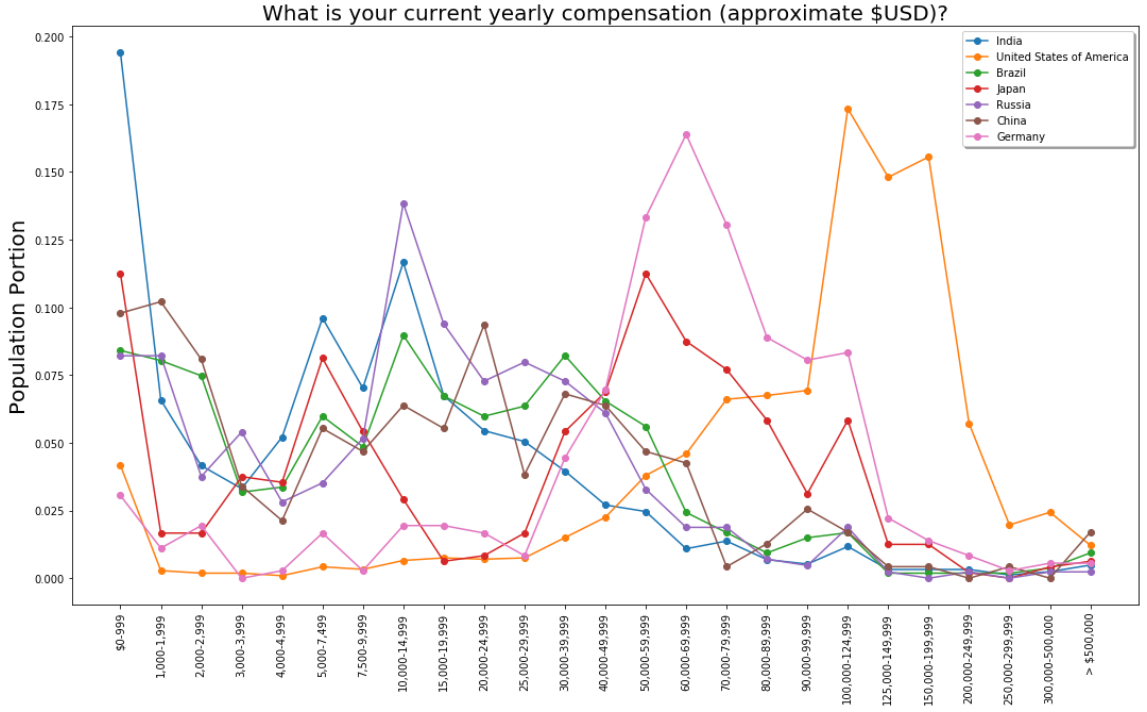
biggest contributing factors?

Research Methods:

The way that the salary data is organized is in categories. Because there are more than 20 categories, it would be difficult to have an accurate model. First thing we'll do is take our mc_interest frame from section 1 and turn the salary data into 5 categories. We will then use the decision trees model to train and test for salary prediction.

In [43]:

```
#Recall salary categories from section 2
column = top_10['Q10']
num_countries = 7
category_df = top_10_category(column)
category_df = category_df.reindex([0, 1, 7, 12, 15, 17, 20, 2, 5, 8,
category_df = category_df.iloc[:, :(num_countries+1)]
colors = ['red', 'blue', 'green', 'yellow', 'purple', 'black', 'pink'
columns = category_df.columns
fig, ax = plt.subplots(figsize = (18, 10))
ax.set_title(q_dict[column.name], fontsize = 20)
ax.set_ylabel('Population Portion', fontsize = 20)
plt.xticks(rotation = 'vertical')
for i in range(1, len(columns)):
    ax.plot(category_df.iloc[:,0], category_df.iloc[:,i], marker='o')
legend = plt.legend(category_df.iloc[:,1:].columns, loc = 0, shadow=T
plt.show()
```



In [44]:

```
#We want to predict the salary, so let's drop those pesky 'nan' value
salary_data = mc_interest.dropna(subset=['Q10'])

#Filter for 5 categories
categories = salary_data['Q10'].unique()
categories.sort()
new_order = [0, 1, 7, 12, 15, 17, 20, 2, 5, 8, 10, 13, 16, 18, 19, 21
cat_list = []
for num in new_order:
    cat_list.append(categories[num])
categories = cat_list
categories
#here's our sorted list of categories, let's narrow it down.
```

Out[44]:

```
['$0-999',
 '1,000-1,999',
 '2,000-2,999',
 '3,000-3,999',
 '4,000-4,999',
 '5,000-7,499',
 '7,500-9,999',
 '10,000-14,999',
 '15,000-19,999',
 '20,000-24,999',
 '25,000-29,999',
 '30,000-39,999',
 '40,000-49,999',
 '50,000-59,999',
 '60,000-69,999',
 '70,000-79,999',
 '80,000-89,999',
 '90,000-99,999',
 '100,000-124,999',
 '125,000-149,999',
 '150,000-199,999',
 '200,000-249,999',
 '250,000-299,999',
```

```
    '300,000-500,000',
    '> $500,000']
```

In [45]:

```
#lets make a dict with old and new values
dict_new_salary = {}
for salary in categories[0:5]:
    dict_new_salary[salary] = '0-4,999'
for salary in categories[5:10]:
    dict_new_salary[salary] = '5,000-24,999'
for salary in categories[10:15]:
    dict_new_salary[salary] = '25,0000-69,999'
for salary in categories[15:20]:
    dict_new_salary[salary] = '70,000-149,999'
for salary in categories[20:25]:
    dict_new_salary[salary] = '150,000+'
dict_new_salary
```

Out[45]:

```
{'$0-999': '0-4,999',
 '1,000-1,999': '0-4,999',
 '2,000-2,999': '0-4,999',
 '3,000-3,999': '0-4,999',
 '4,000-4,999': '0-4,999',
 '5,000-7,499': '5,000-24,999',
 '7,500-9,999': '5,000-24,999',
 '10,000-14,999': '5,000-24,999',
 '15,000-19,999': '5,000-24,999',
 '20,000-24,999': '5,000-24,999',
 '25,000-29,999': '25,0000-69,999',
 '30,000-39,999': '25,0000-69,999',
 '40,000-49,999': '25,0000-69,999',
 '50,000-59,999': '25,0000-69,999',
 '60,000-69,999': '25,0000-69,999',
 '70,000-79,999': '70,000-149,999',
 '80,000-89,999': '70,000-149,999',
 '90,000-99,999': '70,000-149,999',
 '100,000-124,999': '70,000-149,999',
 '125,000-149,999': '70,000-149,999',
 '150,000-199,999': '150,000+',
 '200,000-249,999': '150,000+',
 '250,000-299,999': '150,000+',
```

```
 '300,000-500,000': '150,000+',
 '> $500,000': '150,000+'}
```

In [46]:

```
#Now we can use this dict to rename the categories in the salary_data
salary_data = salary_data.replace({'Q10':dict_new_salary})
salary_data['Q10'].unique()
```

Out[46]:

```
array(['25,0000-69,999', '5,000-24,999', '150,000+',
'0-4,999',
       '70,000-149,999'], dtype=object)
```

In [47]:

```
#Let's use our ML method
y = salary_data['Q10'].copy()
X = salary_data[['Q2', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q14
```

In [48]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0
salary_classifier = DecisionTreeClassifier(max_leaf_nodes=12, random_
salary_classifier.fit(X_train, y_train)
predictions = humidity_classifier.predict(X_test)
accuracy_score(y_true = y_test, y_pred = predictions)
```

```
---------------------------------------------------------
---------------------
ValueError                                Traceback (m
ost recent call last)
<ipython-input-48-fa3caebd6e27> in <module>
      1 X_train, X_test, y_train, y_test = train_test_
split(X, y, test_size=0.20, random_state=None)
      2 salary_classifier = DecisionTreeClassifier(max
_leaf_nodes=12, random_state=0)
----> 3 salary_classifier.fit(X_train, y_train)
      4 predictions = humidity_classifier.predict(X_te
st)
      5 accuracy_score(y_true = y_test, y_pred = predi
ctions)

~\Anaconda3\lib\site-packages\sklearn\tree\tree.py in
fit(self, X, y, sample_weight, check_input, X_idx_sort
ed)
    814                 sample_weight=sample_weight,
    815                 check_input=check_input,
--> 816                 X_idx_sorted=X_idx_sorted)
    817         return self
    818

~\Anaconda3\lib\site-packages\sklearn\tree\tree.py in
fit(self, X, y, sample_weight, check_input, X_idx_sort
ed)
    128         random_state = check_random_state(self
.random_state)
    129         if check_input:
--> 130             X = check_array(X, dtype=DTYPE, ac
cept_sparse="csc")
    131             y = check_array(y, ensure_2d=False
, dtype=None)
```

```
    132             if issparse(X):

~\Anaconda3\lib\site-packages\sklearn\utils\validatio
n.py in check_array(array, accept_sparse, accept_large
_sparse, dtype, order, copy, force_all_finite, ensure_
2d, allow_nd, ensure_min_samples, ensure_min_features,
warn_on_dtype, estimator)
    494             try:
    495                 warnings.simplefilter('error',
ComplexWarning)
--> 496                 array = np.asarray(array, dtyp
e=dtype, order=order)
    497             except ComplexWarning:
    498                 raise ValueError("Complex data
not supported\n"

~\Anaconda3\lib\site-packages\numpy\core\numeric.py in
asarray(a, dtype, order)
    536
    537     """
--> 538     return array(a, dtype, copy=False, order=o
rder)
    539
    540

ValueError: could not convert string to float: 'Male'
```

Conclusion

We have yet to learn a ML method to make decisions, cluster, or create regressions based on categorical data. We can use numerical data to predict categorical data, but we still must learn how to use categorical data to predict categorical data. It will be fun to return to this question after having taken the 'Machine Learning' EdX course.