Libraries

In [1]:
```python
import os
import copy
import datetime
import warnings

import random
from datetime import datetime
import re

import numpy as np
from scipy.stats import t
import pandas as pd
import keras

from matplotlib import pyplot as plt
import matplotlib as mpl

from wordcloud import WordCloud

import seaborn as sns
```

In [2]:
```python
DATA_PATH = '../input/data-science-for-good-careervillage'
np.random.seed(42)

pd.set_option('display.max_columns', 100, 'display.width', 1024)
pd.options.mode.chained_assignment = None
warnings.filterwarnings('ignore')

mpl.rcParams['axes.facecolor'] = '#12162f'
plt.grid(False)
plt.close()
```

In [3]:
```python
# Read CSV casitng dates
answers = pd.read_csv(os.path.join(DATA_PATH, 'answers.csv'))
answer_scores = pd.read_csv(os.path.join(DATA_PATH, 'answer_scores.csv'))
comments = pd.read_csv(os.path.join(DATA_PATH, 'comments.csv'))
emails = pd.read_csv(os.path.join(DATA_PATH, 'emails.csv'))
groups = pd.read_csv(os.path.join(DATA_PATH, 'groups.csv'))
group_memberships = pd.read_csv(os.path.join(DATA_PATH, 'group_memberships.csv'))
matches = pd.read_csv(os.path.join(DATA_PATH, 'matches.csv'))
professionals = pd.read_csv(os.path.join(DATA_PATH, 'professionals.csv'))
questions = pd.read_csv(os.path.join(DATA_PATH, 'questions.csv'))
question_scores = pd.read_csv(os.path.join(DATA_PATH, 'question_scores.csv'))
school_memberships = pd.read_csv(os.path.join(DATA_PATH, 'school_memberships.csv'))
students = pd.read_csv(os.path.join(DATA_PATH, 'students.csv'))
tags = pd.read_csv(os.path.join(DATA_PATH, 'tags.csv'))
tag_questions = pd.read_csv(os.path.join(DATA_PATH, 'tag_questions.csv'))
tag_users = pd.read_csv(os.path.join(DATA_PATH, 'tag_users.csv'))
```

In [4]:
```python
print('Important numbers:')
print('\nThere are:')
print(f'- {len(students)} Students.', end="\t")
print(f'- {len(professionals)} Professionals.')
print(f'- {len(questions)} Questions.', end="\t")
print(f'- {len(answers)} Answers.')
print(f'- {len(tags)} Tags.', end="\t\t")
print(f'- {len(comments)} Comments.')
print(f'- {school_memberships["school_memberships_school_id"].nunique()} Schools.', end="\t\t")
print(f'- {len(groups)} Groups.')
print(f'- {len(emails)} Emails were sent.')
```

```
Important numbers:

There are:
- 30971 Students.        - 28152 Professionals.
- 23931 Questions.       - 51123 Answers.
- 16269 Tags.            - 14966 Comments.
- 2706 Schools.          - 49 Groups.
- 1850101 Emails were sent.
```

Mô t

23,931 câu hỏi từ 12,329 học sinh

51,123 câu trả lời cho 23110 câu hỏi từ 10169 chuyên gia

In [5]:
```python
questions.describe()
```

Out[5]:

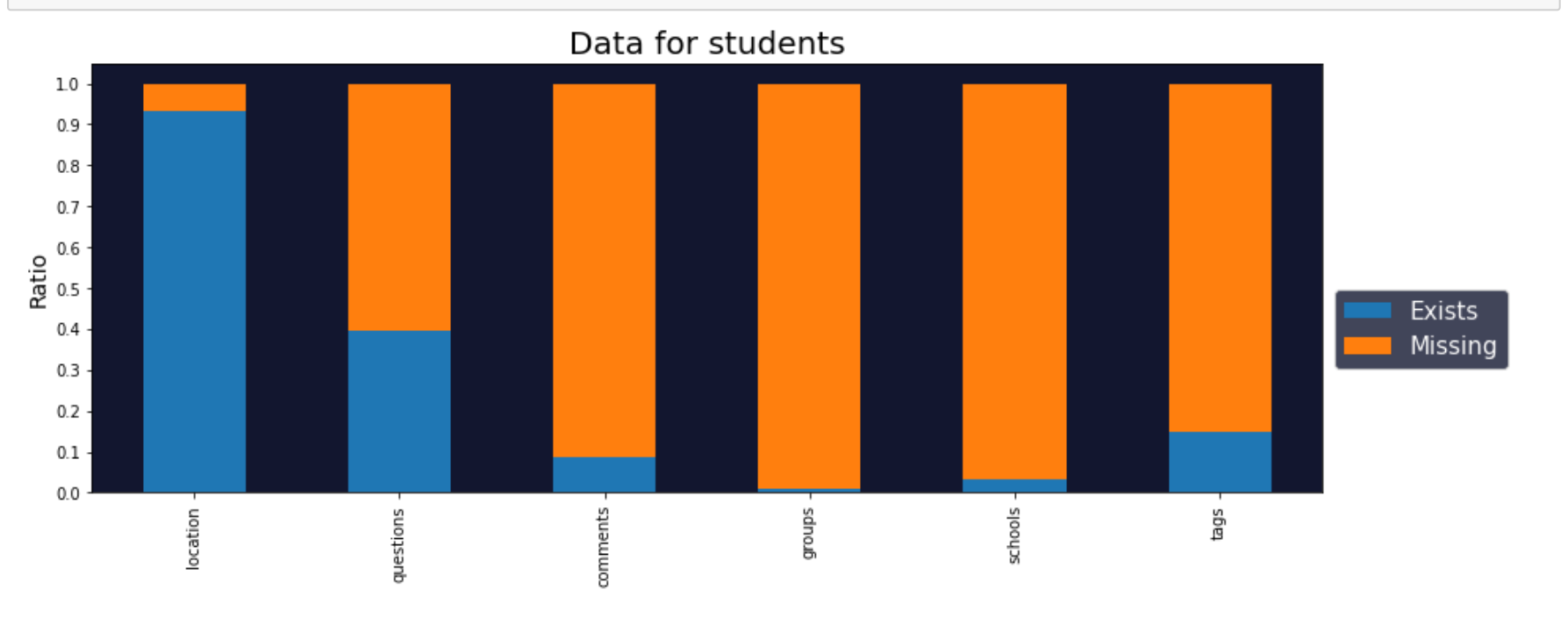
	questions_id	questions_author_id	questions_date_added	questions_title	questions_body
count	23931	23931	23931	23931	23931
unique	23931	12329	23869	23739	23681
top	0b8360444f49406195078e068b3173b2	2fe767de78fa4dfd83f0021cf7712064	2018-04-06 17:48:33 UTC+0000	How do I become an IAS officer?	Accounting student at Towson university #acco...
freq	1	93	2	6	16

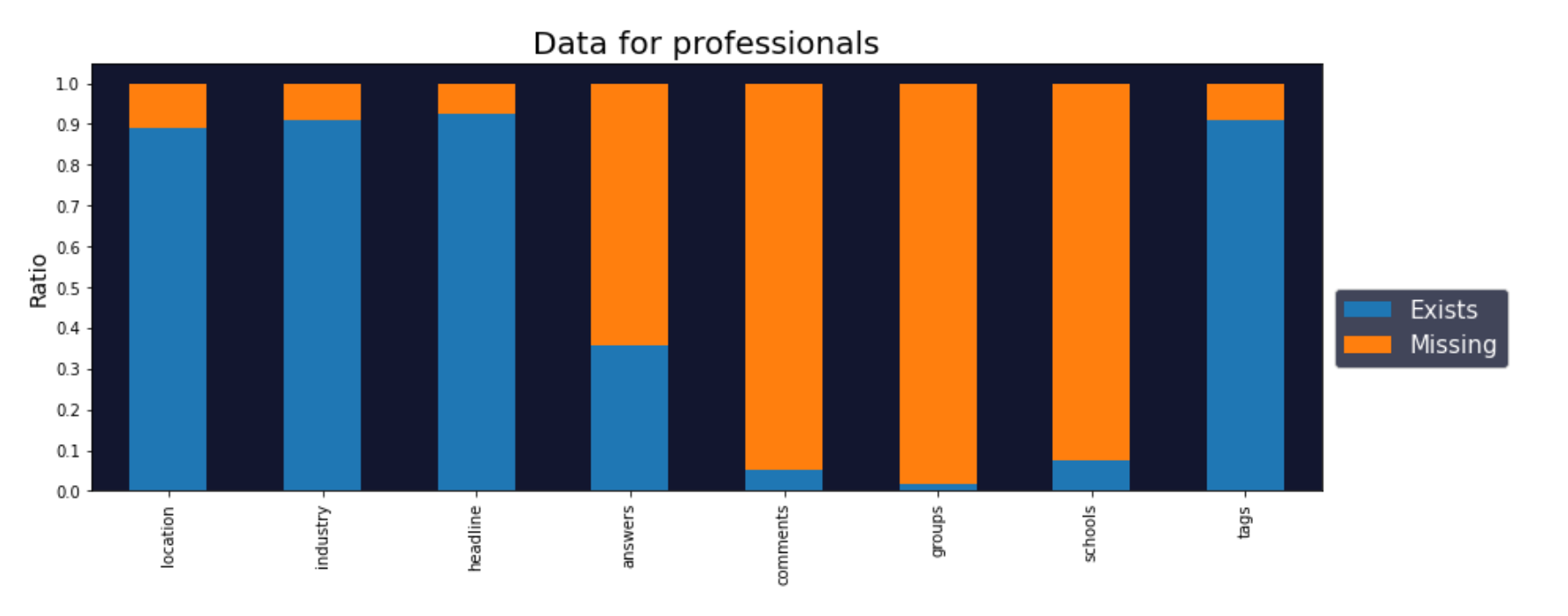
In [6]:
```python
answers.describe()
```

Out[6]:

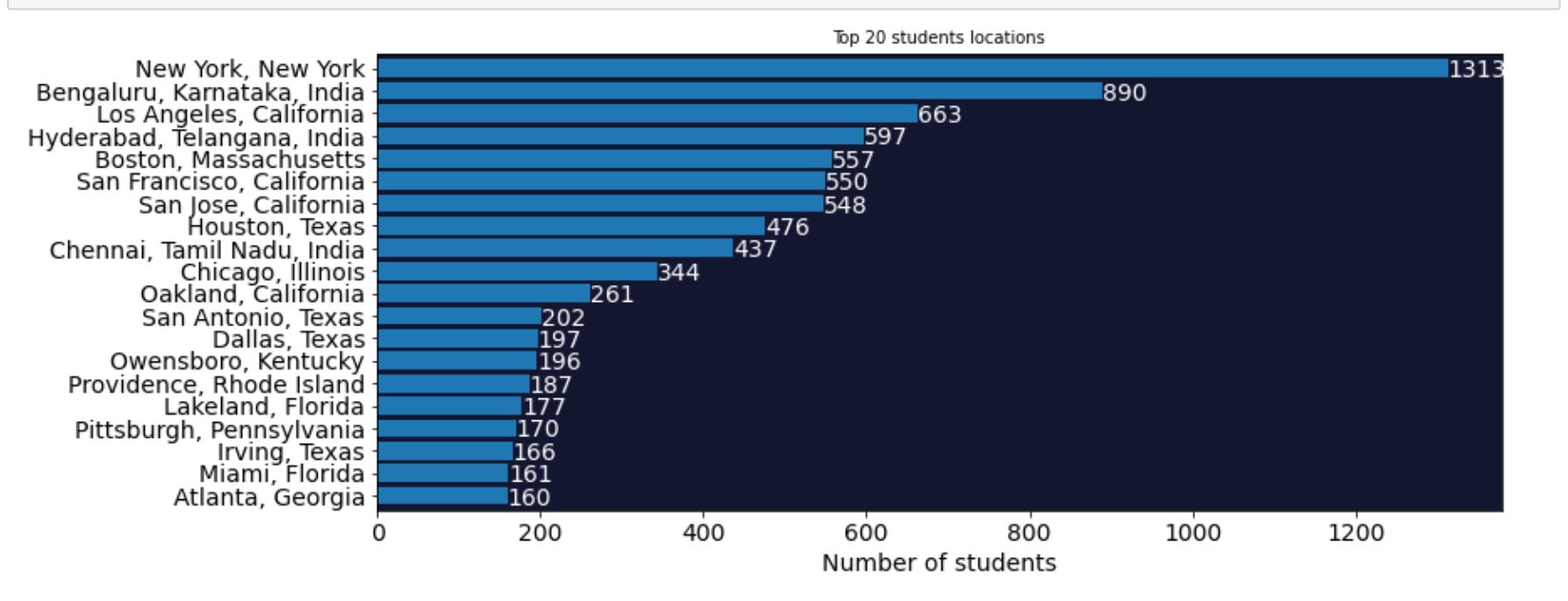
	answers_id	answers_author_id	answers_question_id	answers_date_added	ar
count	51123	51123	51123	51123	
unique	51123	10169	23110	51062	
top	48b662bb26684cf2a9f70a3f6d9414a1	36ff3b3666df400f956f8335cf53e09e	8eb6ba7af57846acbfec5633e537192a	2017-05-10 18:39:51 UTC+0000	style= colo
freq	1	1710	58	2	

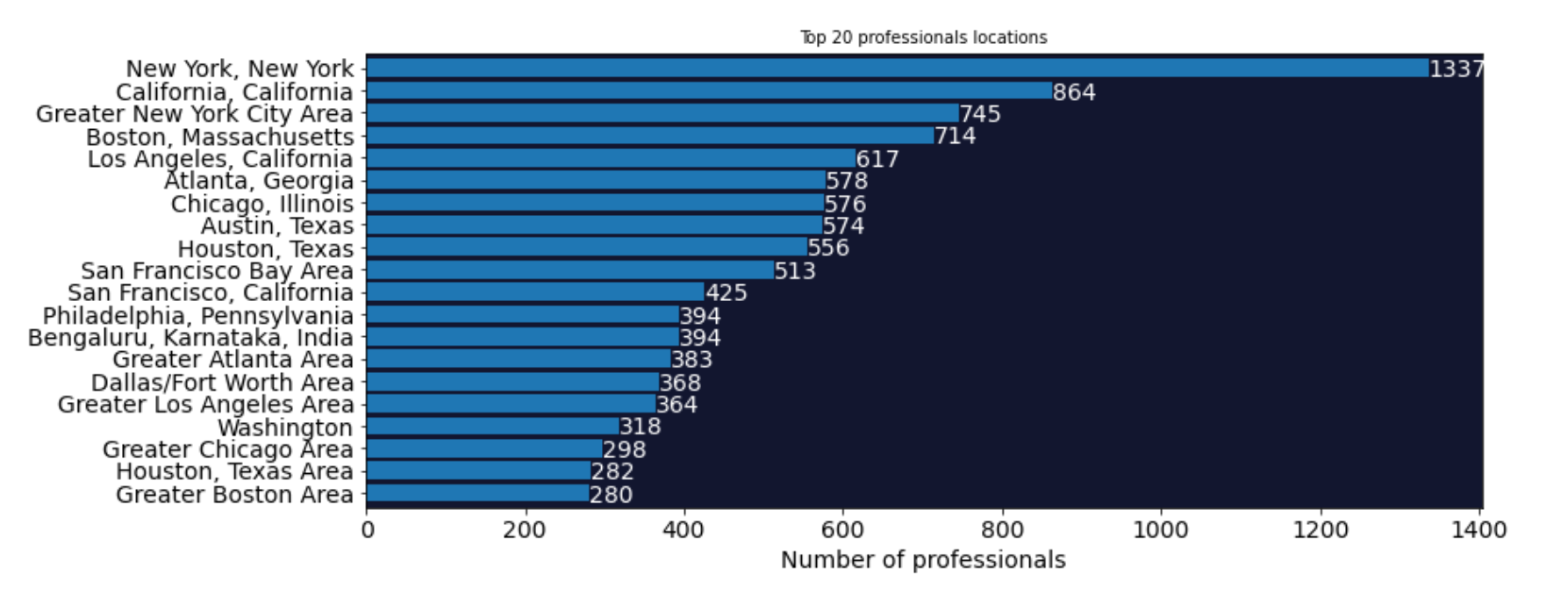
In [7]:
```python
exists = 'Exists'
miss = 'Missing'

fields = {
    'students': {
        'df': students,
        'features': ['location'],
        'count': {
            'questions': questions,
            'comments': comments
        },
        'memberships': {
            'groups': group_memberships,
            'schools': school_memberships
        },
        'plts': {}
    },
    'professionals': {
        'df': professionals,
        'features': ['location', 'industry', 'headline'],
        'count': {
            'answers': answers,
            'comments': comments
        },
        'memberships': {
            'groups': group_memberships,
            'schools': school_memberships
        },
        'plts': {}
    }
}

for user in fields.keys():
    user_spec = fields[user]

    df = user_spec['df']

    tmp = df[['{}_{}'.format(user, feature) for feature in user_spec['features']]]
    tmp = tmp.fillna(miss)
    tmp[tmp != miss] = exists

    for feature in user_spec['features']:
        user_spec['plts'][feature] = tmp.groupby('{}_{}'.format(user, feature)).size()/len(tmp)

    # Counts
    for key, cdf in user_spec['count'].items():
        tf = cdf.groupby('{}_author_id'.format(key)).size()
        tf['counts'] = pd.merge(df, pd.DataFrame(tf.rename('count')), left_on='{}_id'.format(user), rig
ht_index=True, how='left')['count'].fillna(0).astype(int)

        tf = tf['counts']
        tf[tf > 0] = exists
        tf[tf != exists] = miss

        user_spec['plts'][key] = tf.groupby(tf.values).size()/len(tf)

    # Counts
    for key, mdf in user_spec['memberships'].items():
        unique_userid_with_membership = mdf['{}_memberships_user_id'.format(key[:-1])].unique()
        tf = pd.DataFrame()
        tf['val'] = df['{}_id'.format(user)]

        tf[tf['val'].isin(unique_userid_with_membership)] = exists
        tf[tf['val'] != exists] = miss

        user_spec['plts'][key] = tf.groupby('val').size()/len(tf)

    # Tags
    unique_user_ids_with_tags = tag_users['tag_users_user_id'].unique()
    tf = pd.DataFrame()
    tf['val'] = df['{}_id'.format(user)]

    tf[tf['val'].isin(unique_user_ids_with_tags)] = exists
    tf[tf['val'] != exists] = miss

    user_spec['plts']['tags'] = tf.groupby('val').size()/len(tf)
    plt_data = pd.DataFrame(user_spec['plts'])

    plt_data.T.plot(kind='bar', stacked=True, figsize=(14, 5))
    plt.ylabel('Ratio', fontsize=14)
    plt.title('Data for {}'.format(user), fontsize=20)
    plt.yticks(np.arange(0, 1.05, 0.1))
    leg = plt.legend(bbox_to_anchor=(1, 0.5), fontsize=15)
    for text in leg.get_texts():
        plt.setp(text, color = 'w')
    plt.show()
```



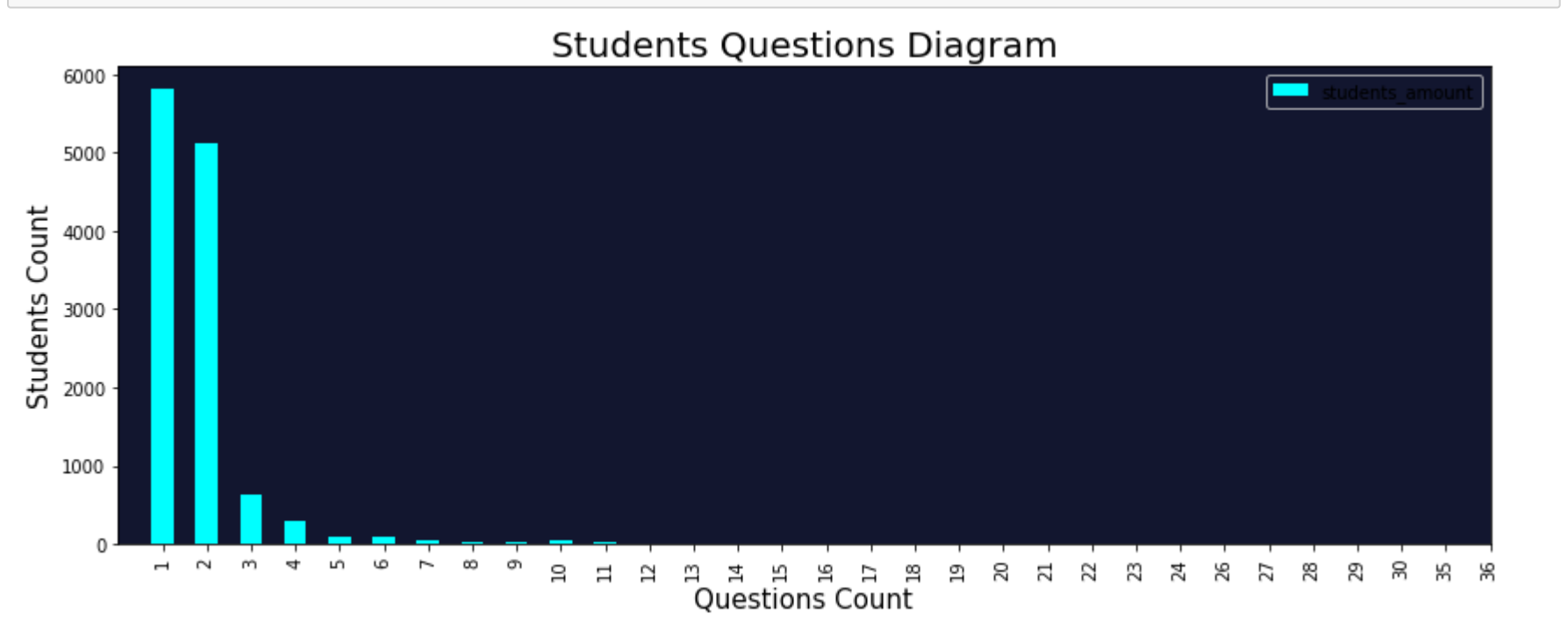


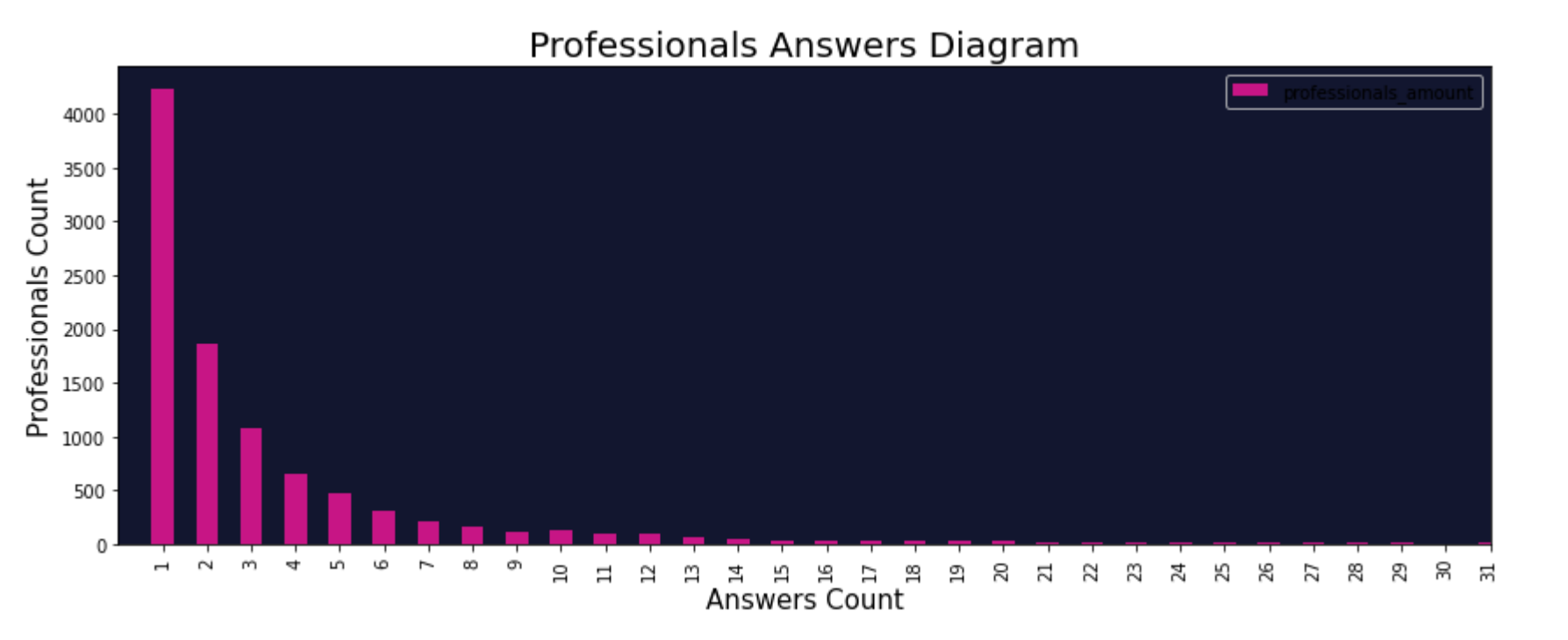
In [8]:
```python
n_locations = 20

users = [
    ('students', students),
    ('professionals', professionals)
]

for user, df in users:
    locations = df['{}_location'.format(user)].value_counts().sort_values(ascending=True).tail(n_locati
ons)

    ax = locations.plot(kind='barh',figsize=(12, 5),width=0.8, fontsize=14)
    ax.set_title('Top %s {} locations'.format(user) % n_locations, fontsize=10)
    ax.set_xlabel('Number of {}'.format(user), fontsize=14)
    for p in ax.patches:
        ax.annotate(str(p.get_width()), (p.get_width(), p.get_y()), color='w', fontsize=14)
    plt.show()
```



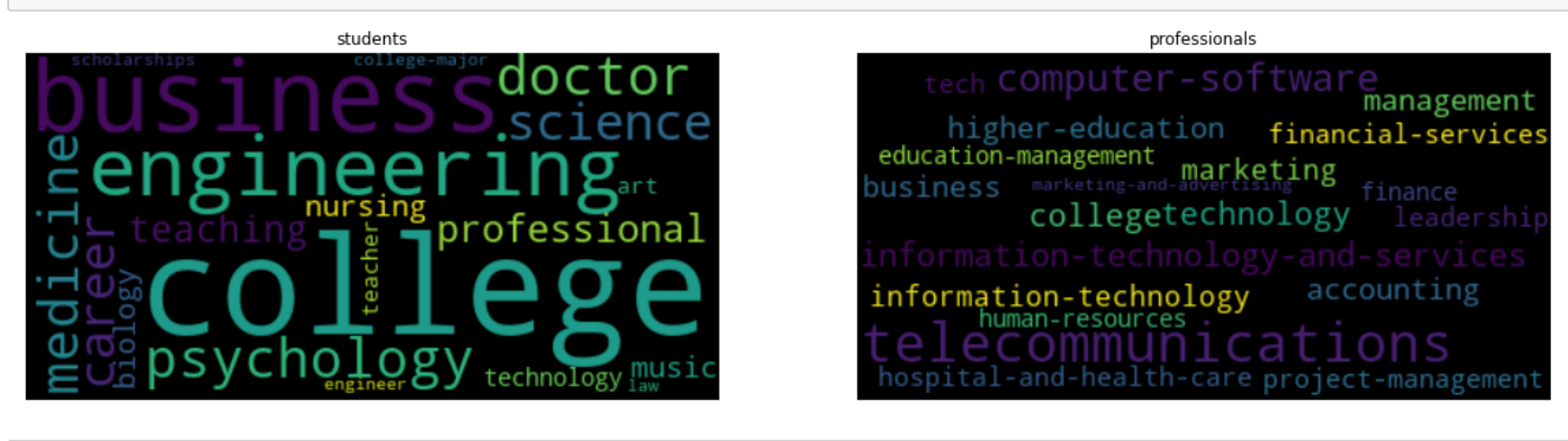


In [9]:
```python
from collections import Counter

ll = [
    ('students', 'questions', students, questions),
    ('professionals', 'answers', professionals, answers)
]

colors = {'students' : 'cyan', 'professionals' : 'mediumvioletred'}

for user, entity, user_df, entity_df in ll:
    tm = dict(sorted(Counter(pd.merge(user_df, entity_df, left_on='{}_id'.format(user), right_on='{}_au
thor_id'.format(entity), how='inner').groupby('{}_id'.format(user)).size().values).items()))
    t_d = {}
    t_d['{}_amount'.format(entity)] = list(tm.keys())
    t_d['{}_amount'.format(user)] = list(tm.values())

    plt_data = pd.DataFrame(t_d)

    plt_data.plot(x='{}_amount'.format(entity), y='{}_amount'.format(user), kind='bar', figsize=(14, 5
), color=colors[user])
    plt.xlim(-1, 30)
    plt.xlabel('{} Count'.format(entity.capitalize()), fontsize=15)
    plt.ylabel('{} Count'.format(user.capitalize()), fontsize=15)
    plt.title('{} {} Diagram'.format(user.capitalize(), entity.capitalize()), fontsize=20)
    plt.show()
```





In [10]:
```python
entities = [
    ('students', students),
    ('professionals', professionals),
    ('questions', questions)
]

dfs = []

for entity, df in entities:
    if entity == 'questions':
        df = tag_questions
        df = pd.merge(df, tags, left_on='tag_questions_tag_id', right_on='tags_tag_id')
    else:
        df = tag_users[tag_users['tag_users_user_id'].isin(df['{}_id'.format(entity)])]
        df = pd.merge(df, tags, left_on='tag_users_tag_id', right_on='tags_tag_id')

    df['entity_type'] = entity

    dfs.append(df)


plt_data = pd.concat(dfs)


plt_data = plt_data[['tags_tag_name', 'entity_type']].pivot_table(index='tags_tag_name', columns='entit
y_type', aggfunc=len, fill_value=0)

for entity, df in entities:
    plt_data[entity] = plt_data[entity] / len(df)

plt_data['sum'] = (plt_data['professionals'] + plt_data['students'] + plt_data['questions'])
plt_data = plt_data.sort_values(by='sum', ascending=False).drop(['sum'], axis=1).head(100)


# Wordcloud
plt.figure(figsize=(20, 20))
wordloud_values = ['students', 'professionals']
axisNum = 1
for wordcloud_value in wordloud_values:
    wordcloud = WordCloud(margin=0, max_words=20, random_state=42).generate_from_frequencies(plt_data[w
ordcloud_value])
    ax = plt.subplot(1, 2, axisNum)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(wordcloud_value)
    plt.axis("off")
    axisNum += 1
plt.show()
```



In [13]:
```python
dfs
```

Out[13]:
```
[       tag_users_tag_id                 tag_users_user_id  tags_tag_id              tags_tag_name e
ntity_type
 0                 8593  82cc74c0e18347e4b68b219e250eb10f         8593                  scientist
 students
 1                 8593  f3a5c976bec34f4289aa58938a837300         8593                  scientist
 students
 2                 8593  b3e73d394b19488b9c9ec13fcd7c43a7         8593                  scientist
 students
 3                 8593  59da5285144044c8a883fa9a6d40cd6a         8593                  scientist
 students
 4                 8593  d957ed2aac584dd79e3280ff679e818b         8593                  scientist
 students
 ...                ...                               ...          ...                        ...
      ...
 19200            38991  f6361bef40324e7e8af069459f1ba6d1        38991  #howtogetinsidetheindustry
 students
 19201            31967  f6361bef40324e7e8af069459f1ba6d1        31967                   #masters
 students
 19202            38992  f6361bef40324e7e8af069459f1ba6d1        38992               aplacetostart
 students
 19203            39007  936ef033ccbd47f3a37e90bd52fdb8f9        39007                  electrian
 students
 19204            11774  7032b7f09e6540ce99af0c56fedbbf4f        11774              womens-studies
 students

 [19205 rows x 5 columns],
         tag_users_tag_id                 tag_users_user_id  tags_tag_id           tags_tag_name    e
ntity_type
 0                    593  c72ab38e073246e88da7e9a4ec7a4472          593       computer-software  pro
fessionals
 1                    593  8db519781ec24f2e8bdc67c2ac53f614          593       computer-software  pro
fessionals
 2                    593  9ab6b54d55b24299a4795584508db4ff          593       computer-software  pro
fessionals
 3                    593  e327399c48584fcf81e433828a6d8715          593       computer-software  pro
fessionals
 4                    593  92494d9dc2124507972c5306badc6727          593       computer-software  pro
fessionals
 ...                  ...                               ...          ...                     ...
      ...
 117453             31813  f720e8d8e76c4e26b6051609a13c9ca5        31813     #customerexperience  pro
fessionals
 117454             39260  4a7e1e5dd884488da283e1cab6ad11e0        39260    #inbuildingsolutions  pro
fessionals
 117455             38017  2129756263854d86a7f56d6bf1172f70        38017            advanced-math  pro
fessionals
 117456             39261  2129756263854d86a7f56d6bf1172f70        39261             mathmetician  pro
fessionals
 117457             39230  0e98d45e17b249cb9a4614dd644f4813        39230  #ignitingpossibilities  pro
fessionals

 [117458 rows x 5 columns],
        tag_questions_tag_id         tag_questions_question_id  tags_tag_id tags_tag_name entity_type
 0                     28930  cb43ebee01364c68ac61d347a393ae39        28930         minor   questions
 1                     28930  47f55e85ce944242a5a347ab85a8ffb4        28930         minor   questions
 2                     28930  ccc30a033a0f4dfdb2eb987012f25792        28930         minor   questions
 3                     28930  e30b274e48d741f7bf50eb5e7171a3c0        28930         minor   questions
 4                     28930  3d22742052df4989b311b4195cbb0f1a        28930         minor   questions
 ...                     ...                               ...          ...           ...         ...
 76548                 39183  72affe86909d49c09761a1d376476c92        39183       austria   questions
 76549                 39248  5b86153f2b6c46a4801bb39e10e01281        39248  weldernation   questions
 76550                 39249  0946a81d511b48c58376fd59058d3b12        39249        welder   questions
 76551                 39249  b4de8ccfcd874481bfc80ceacb0caac8        39249        welder   questions
 76552                 39250  0946a81d511b48c58376fd59058d3b12        39250        nation   questions

 [76553 rows x 5 columns]]
```