**Yelp Reviews: Text Analysis**

In [38]:

**import** **pandas** **as** **pd**  
**import** **matplotlib.pyplot** **as** **plt**  
**import** **seaborn** **as** **sns**  
%**matplotlib** inline

**Loading CSV into Pandas Dataframe**[**¶**](#gjdgxs)

In [2]:

dataframe = pd.read\_csv("yelp\_reviews.csv")

Dropping below Columns as these columns are not useful for Exploratory Analysis and can be used in Rating Prediction.

##### ***Columns dropped : Rating, Text, time\_created, url, user***[***¶***](#30j0zll)

In [23]:

physicians = dataframe.drop(["rating", "text", "time\_created", "url", "user"], axis=1).drop\_duplicates()

In [24]:

physicians.head()

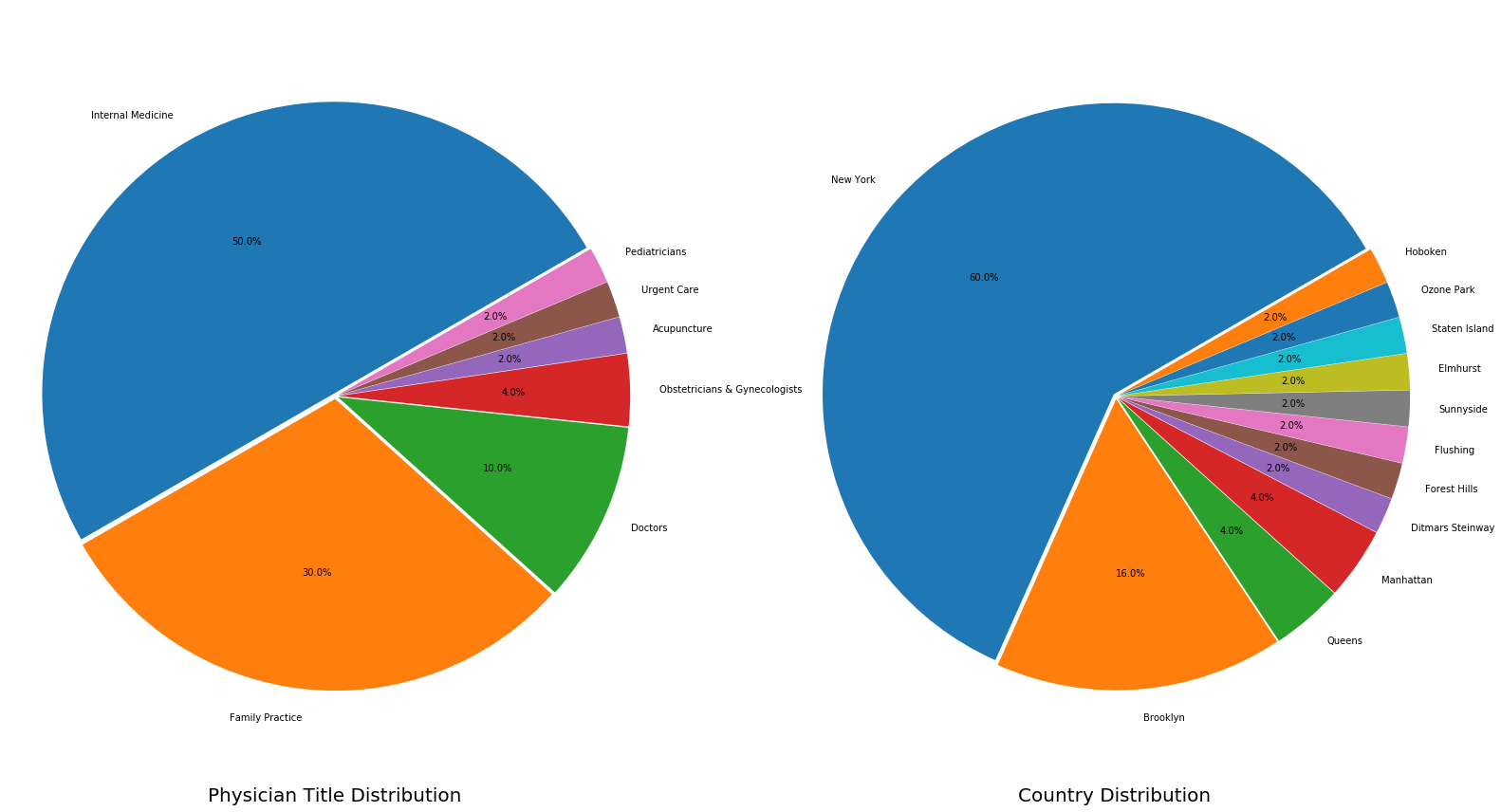
Out[24]:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **address** | **alias** | **avg\_rating** | **city** | **country** | **physician** | **state** | **title** | **zip\_code** |
| **0** | 35 E 21st St, Fl 7, New York, NY 10010 | internalmed | 4.5 | New York | US | Laura Guderian, MD | NY | Internal Medicine | 10010 |
| **3** | 14 E 4th St, Ste 501, New York, NY 10012 | internalmed | 4.5 | New York | US | Mohawk Village Medical | NY | Internal Medicine | 10012 |
| **6** | 111 Broadway, Ste 1302, New York, NY 10006 | familydr | 4.5 | New York | US | Michael Raffinan, MD | NY | Family Practice | 10006 |
| **9** | 476 7th St, Brooklyn, NY 11215 | internalmed | 5.0 | Brooklyn | US | Inna Grishina, MD | NY | Internal Medicine | 11215 |
| **12** | 136-21 Roosevelt Ave, Ste 409, Flushing, NY 11354 | internalmed | 4.5 | Flushing | US | Xiaoyan Song, MD | NY | Internal Medicine | 11354 |

**Visualizing Physicians Title and City Distribution in Dataset**[**¶**](#1fob9te)

In [111]:

title\_count\_dict = physicians.title.value\_counts().to\_dict()  
labels = title\_count\_dict.keys()  
sizes = title\_count\_dict.values()  
fig1, axes = plt.subplots(1,2,figsize=(27,15))  
explode = [0.01 **for** each **in** sizes]  
axes[0].pie(sizes, labels=labels, autopct='**%1.1f%%**',  
 shadow=**False**, startangle=30, explode=explode)  
axes[0].axis('equal') *# Equal aspect ratio ensures that pie is drawn as a circle.*  
axes[0].set\_xlabel('Physician Title Distribution')  
axes[0].xaxis.label.set\_size(20)  
  
city\_count\_dict = physicians.city.value\_counts().to\_dict()  
labels = city\_count\_dict.keys()  
sizes = city\_count\_dict.values()  
explode = [0.01 **for** each **in** sizes]  
axes[1].pie(sizes, labels=labels, autopct='**%1.1f%%**',  
 shadow=**False**, startangle=30, explode=explode)  
axes[1].axis('equal') *# Equal aspect ratio ensures that pie is drawn as a circle.*  
axes[1].set\_xlabel('Country Distribution')  
axes[1].xaxis.label.set\_size(20)  
  
plt.show()



**Visualizing Ratings Frequency Distribution**[**¶**](#3znysh7)

In [132]:

physicians.avg\_rating.plot(kind="hist")

Out[132]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x206327a9198>

