In [1]:
```
%matplotlib inline
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
```

In [3]:
```
#load data table and display
death_analysis_df = pd.read_csv(r"C:\Users\keert\OneDrive\Documents\machine learning projecs\worldometer_snapshots_April18_to_May18.csv")
death_analysis_df
```

Out[3]:

	Date	Country	Population	Total Tests	Total Cases	Total Deaths	Total Recovered	Serious or Critical	Active Cases
0	2020-04-18	USA	330774664	3722145.0	738792.0	39014.0	68269.0	13551.0	631509.0
1	2020-04-18	Russia	145927292	1831892.0	36793.0	313.0	3057.0	8.0	33423.0
2	2020-04-18	Spain	46752703	930235.0	194416.0	20043.0	74797.0	7371.0	99576.0
3	2020-04-18	Brazil	212380932	62985.0	36722.0	2361.0	14026.0	6634.0	20335.0
4	2020-04-18	UK	67844241	460437.0	114217.0	15464.0	NaN	1559.0	98409.0
...	...	...	...	...	...	...	...	...	...
6598	2020-05-18	St. Barth	9874	NaN	6.0	NaN	6.0	NaN	0.0
6599	2020-05-18	Western Sahara	595462	NaN	6.0	NaN	6.0	NaN	0.0
6600	2020-05-18	Anguilla	14987	NaN	3.0	NaN	3.0	NaN	0.0
6601	2020-05-18	Lesotho	2140235	NaN	1.0	NaN	NaN	NaN	1.0
6602	2020-05-18	Saint Pierre Miquelon	5797	NaN	1.0	NaN	1.0	NaN	0.0

6603 rows × 9 columns

In [4]:
```
#display sub-table of specific country
country_name = 'USA'

country_df = death_analysis_df.loc[death_analysis_df['Country'] == country_name, :].reset_index(drop=True)
country_df
```

Out[4]:

	Date	Country	Population	Total Tests	Total Cases	Total Deaths	Total Recovered	Serious or Critical	Active Cases
0	2020-04-18	USA	330774664	3722145.0	738792.0	39014.0	68269.0	13551.0	631509.0
1	2020-04-19	USA	330774664	3861549.0	763832.0	40553.0	71003.0	13566.0	652276.0
2	2020-04-20	USA	330774664	4026360.0	792759.0	42514.0	72389.0	13951.0	677856.0
3	2020-04-21	USA	330774664	4187392.0	818744.0	45318.0	82923.0	14016.0	690503.0
4	2020-04-22	USA	330774664	4325342.0	848717.0	47659.0	84050.0	14016.0	717008.0
5	2020-04-23	USA	330774664	4696704.0	880204.0	49845.0	85922.0	14997.0	744437.0
6	2020-04-24	USA	330774664	5015602.0	925038.0	52185.0	110432.0	15097.0	762421.0
7	2020-04-25	USA	330774664	5279237.0	960651.0	54256.0	118162.0	15110.0	788233.0
8	2020-04-26	USA	330774664	5470464.0	987160.0	55413.0	118781.0	15143.0	812966.0
9	2020-04-27	USA	330774664	5696928.0	1010356.0	56797.0	138990.0	14186.0	814569.0
10	2020-04-28	USA	330774664	5919847.0	1035765.0	59266.0	142238.0	15298.0	834261.0
11	2020-04-29	USA	330774664	6139911.0	1064194.0	61656.0	147411.0	18671.0	855127.0
12	2020-04-30	USA	330774664	6391887.0	1095023.0	63856.0	152324.0	15226.0	878843.0
13	2020-05-01	USA	330774664	6699878.0	1131030.0	65753.0	161563.0	16481.0	903714.0
14	2020-05-02	USA	330774664	7196740.0	1160774.0	67444.0	178263.0	16139.0	941261.0
15	2020-05-03	USA	330774664	7196740.0	1188122.0	68598.0	178263.0	16139.0	941261.0
16	2020-05-04	USA	330774664	7462431.0	1212835.0	69921.0	180027.0	16050.0	954887.0
17	2020-05-05	USA	330774664	7727811.0	1237633.0	72271.0	200626.0	16179.0	964736.0
18	2020-05-06	USA	330774664	8005435.0	1263092.0	74799.0	212981.0	15827.0	975312.0
19	2020-05-07	USA	330774664	8297562.0	1292623.0	76928.0	217250.0	16995.0	998445.0
20	2020-05-08	USA	330774664	8636435.0	1321785.0	78615.0	223603.0	16978.0	1019567.0
21	2020-05-09	USA	330774664	8918263.0	1347309.0	80037.0	238078.0	16816.0	1029194.0
22	2020-05-10	USA	330774664	9444525.0	1367638.0	80787.0	256336.0	16514.0	1030515.0
23	2020-05-11	USA	330774664	9619855.0	1395834.0	81795.0	262225.0	16484.0	1041814.0
24	2020-05-12	USA	330774664	9935720.0	1408636.0	83425.0	296746.0	16473.0	1028465.0
25	2020-05-13	USA	330774664	10269996.0	1430348.0	85197.0	310259.0	16349.0	1034892.0
26	2020-05-14	USA	330774664	10638993.0	1457593.0	86912.0	318027.0	16240.0	1052654.0
27	2020-05-15	USA	330774664	11090900.0	1484285.0	88507.0	327751.0	16139.0	1068027.0
28	2020-05-16	USA	330774664	11949625.0	1507773.0	90113.0	339232.0	16248.0	1078428.0
29	2020-05-17	USA	330774664	11875580.0	1527664.0	90978.0	346389.0	16355.0	1090297.0
30	2020-05-18	USA	330774664	12300744.0	1550294.0	91981.0	356383.0	16868.0	1101930.0

In [5]:
```
#sub table of specific date
specific_date = datetime.strptime('16/05/2020', '%d/%m/%Y')

specific_date_df = death_analysis_df.loc[death_analysis_df['Date'] == specific_date.strftime('%Y-%m-%d'), :].reset_index(drop=True)
specific_date_df
```

Out[5]:

	Date	Country	Population	Total Tests	Total Cases	Total Deaths	Total Recovered	Serious or Critical	Active Cases
0	2020-05-16	USA	330774664	11949625.0	1507773.0	90113.0	339232.0	16248.0	1078428.0
1	2020-05-16	Russia	145927292	6656340.0	272043.0	2537.0	63166.0	2300.0	206340.0
2	2020-05-16	Spain	46752703	3037840.0	276505.0	27563.0	192253.0	1208.0	56689.0
3	2020-05-16	Brazil	212380932	735224.0	233142.0	15633.0	89672.0	8318.0	127837.0
4	2020-05-16	UK	67844241	2489563.0	240161.0	34466.0	NaN	1559.0	NaN
...	...	...	...	...	...	...	...	...	...
208	2020-05-16	St. Barth	9874	NaN	6.0	NaN	6.0	NaN	0.0
209	2020-05-16	Western Sahara	595462	NaN	6.0	NaN	6.0	NaN	0.0
210	2020-05-16	Anguilla	14987	NaN	3.0	NaN	3.0	NaN	0.0
211	2020-05-16	Lesotho	2140235	NaN	1.0	NaN	NaN	NaN	1.0
212	2020-05-16	Saint Pierre Miquelon	5797	NaN	1.0	NaN	1.0	NaN	0.0

213 rows × 9 columns

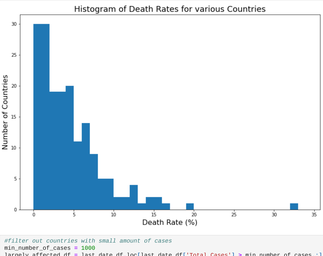
In [6]:
```
#with last date
last_date = datetime.strptime('18/05/2020', '%d/%m/%Y')
last_date_df = death_analysis_df.loc[death_analysis_df['Date'] == last_date.strftime('%Y-%m-%d'), :].reset_index(drop=True)
last_date_df
```

Out[6]:

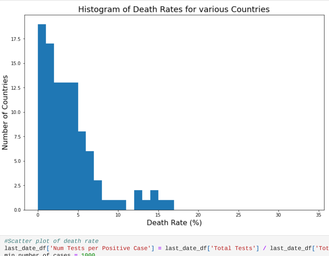
	Date	Country	Population	Total Tests	Total Cases	Total Deaths	Total Recovered	Serious or Critical	Active Cases
0	2020-05-18	USA	330774664	12300744.0	1550294.0	91981.0	356383.0	16868.0	1101930.0
1	2020-05-18	Russia	145927292	7147014.0	290678.0	2722.0	70209.0	2300.0	217747.0
2	2020-05-18	Spain	46752703	3037840.0	278188.0	27709.0	196958.0	1152.0	53521.0
3	2020-05-18	Brazil	212380932	735224.0	255368.0	16853.0	100459.0	8318.0	138056.0
4	2020-05-18	UK	67844241	2682716.0	246406.0	34796.0	NaN	1559.0	NaN
...	...	...	...	...	...	...	...	...	...
208	2020-05-18	St. Barth	9874	NaN	6.0	NaN	6.0	NaN	0.0
209	2020-05-18	Western Sahara	595462	NaN	6.0	NaN	6.0	NaN	0.0
210	2020-05-18	Anguilla	14987	NaN	3.0	NaN	3.0	NaN	0.0
211	2020-05-18	Lesotho	2140235	NaN	1.0	NaN	NaN	NaN	1.0
212	2020-05-18	Saint Pierre Miquelon	5797	NaN	1.0	NaN	1.0	NaN	0.0

213 rows × 9 columns

In [7]:
```
#movie death rate for each country
last_date_df['Case Fatality Ratio'] = last_date_df['Total Deaths'] / last_date_df['Total Cases']

plt.figure(figsize=(12,8))
plt.hist(100 * np.array(last_date_df['Case Fatality Ratio']), bins=np.arange(35))
plt.xlabel('Death Rate (%)', fontsize = 16)
plt.ylabel('Number of Countries', fontsize = 16)
plt.title('Histogram of Death Rates for various Countries', fontsize = 18)
plt.show()
```

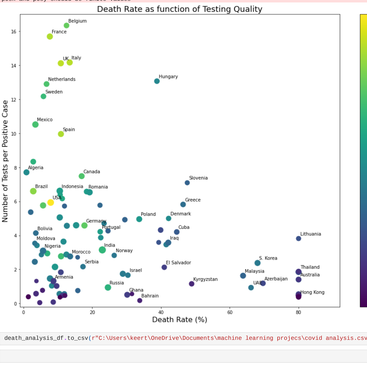


In [8]:
```
#filter out countries with small amount of cases
min_number_of_cases = 1000
largely_affected_df = last_date_df.loc[last_date_df['Total Cases'] > min_number_of_cases, :]

plt.figure(figsize = (12, 8))
plt.hist(100 * np.array(largely_affected_df['Case Fatality Ratio']), bins=np.arange(35))
plt.xlabel('Death Rate (%)', fontsize = 16)
plt.ylabel('Number of Countries', fontsize = 16)
plt.title('Histogram of Death Rates for various Countries', fontsize = 18)
plt.show()
```



In [16]:
```
#Scatter plot of death rate
last_date_df['Num Tests per Positive Case'] = last_date_df['Total Tests'] / last_date_df['Total Cases']
min_number_of_cases = 1000
largely_affected_df = last_date_df.loc[last_date_df['Total Cases'] > min_number_of_cases, :]

x_axis_limit = 80

death_rate_percent = 100 * np.array(largely_affected_df['Case Fatality Ratio'])
num_test_per_positive = np.array(largely_affected_df['Num Tests per Positive Case'])
num_test_per_positive[num_test_per_positive > x_axis_limit] = x_axis_limit
total_num_deaths = np.array(largely_affected_df['Total Deaths'])
population = np.array(largely_affected_df['Population'])

plt.figure(figsize = (16, 12))
plt.scatter(x = num_test_per_positive, y = death_rate_percent,
            s = 0.5 * np.power(np.log(1 + population), 2),
            c = np.log10(1 + total_num_deaths))
plt.colorbar()
plt.xlabel('Death Rate (%)', fontsize = 16)
plt.ylabel('Number of Tests per Positive Case', fontsize = 16)
plt.title('Death Rate as function of Testing Quality', fontsize = 18)
plt.xlim(-1, x_axis_limit + 12)
plt.ylim(-0.2, 17)

#specific countries for plotting
#for all
#countries_to_display = largely_affected_df['Country'], unique(). tolist()
countries_to_display = ['USA', 'Russia', 'Spain', 'Brazil', 'UK', 'Italy', 'France',
                        'Germany', 'India', 'Canada', 'Belgium', 'Mexico', 'Netherlands',
                        'Sweden', 'Portugal', "UAE", 'Poland', 'Indonesia', 'Romania',
                        'Israel', 'Thailand', 'Kyrgyzstan',"El Salvador", 'S. Korea',
                        'Denmark', 'Serbia', 'Norway', 'Algeria', 'Bahrain',"Slovenia",
                        'Greece',"Cuba",'Hong Kong','Lithuania', 'Australia', 'Morocco',
                        'Malaysia', 'Nigeria', 'Moldova', 'Ghana', 'Armenia', 'Bolivia',
                        'Iraq', 'Hungary', 'Cameroon', 'Azerbaijan']
for country_name in countries_to_display:
    country_index = largely_affected_df.index[largely_affected_df['Country'] == country_name]
    plt.text(x = num_test_per_positive[country_index] + 0.5,
             y = death_rate_percent[country_index] + 0.2,
             s = country_name, fontsize = 10)
plt.show()
```

```
posx and posy should be finite values
posx and posy should be finite values
```



In [21]:
```
death_analysis_df.to_csv(r"C:\Users\keert\OneDrive\Documents\machine learning projecs\covid analysis.csv")
```

In [ ]:
```
```