COVID-19 Imapct Assessment

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
data_conf = pd.read_csv(r"E:\Data Analytics\Covid-19-Challenge\Resources\time_series_covid19_confirmed_global_iso3_regions.csv")
data_recov = pd.read_csv(r"E:\Data Analytics\Covid-19-Challenge\Resources\time_series_covid19_recovered_global_iso3_regions.csv")
data_deaths = pd.read_csv(r"E:\Data Analytics\Covid-19-Challenge\Resources\time_series_covid19_deaths_global_iso3_regions.csv")
```

In [3]:
```
type(data_conf)
```

Out[3]:
```
pandas.core.frame.DataFrame
```

What is a DataFrame?

In [4]:
```
data = {'A' : [10,20,30], 'B' : [50,60,70]}
index = [1,2,3]
pd.DataFrame(data, index)
```

Out[4]:

	A	B
1	10	50
2	20	60
3	30	70

In [5]:
```
data_conf.head(100)
```

Out[5]:

	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020	1/26/2020	1/27/2020	1/28/2020	...	2/20/2021	2/2
0	#country+name	#geo+lat	#geo+lon	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	
1	Afghanistan	33.93911	67.709953	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	55580.0	5
2	Albania	41.1533	20.1683	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	99062.0	10
3	Algeria	28.0339	1.6596	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	111764.0	11
4	Andorra	42.5063	1.5218	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	10672.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Congo (Kinshasa)	-4.0383	21.7587	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25080.0	2
96	Costa Rica	9.7489	-83.7534	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	201678.0	20
97	Cote d'Ivoire	7.54	-5.5471	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	31914.0	3
98	Croatia	45.1	15.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	239685.0	23
99	Cuba	21.521757	-77.781167	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	43484.0	4

100 rows × 408 columns

Converting data into list

In [6]:
```
L_conf = data_conf.to_numpy().tolist()
L_recov = data_recov.to_numpy().tolist()
L_deaths = data_deaths.to_numpy().tolist()

#L_conf = data_conf.values.tolist()
#L_recov = data_recov.values.tolist()
#L_deaths = data_deaths.values.tolist()
```

Trying to find the location of China in Dataframe

In [7]:
```
data_conf.loc[data_conf['Country/Region'] == 'China']
```

Out[7]:

	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020	1/26/2020	1/27/2020	1/28/2020	...	2/20/2021	2/21/20
59	China	31.8257	117.2264	1.0	9.0	15.0	39.0	60.0	70.0	106.0	...	994.0	994
60	China	40.1824	116.4142	14.0	22.0	36.0	41.0	68.0	80.0	91.0	...	1046.0	1047
61	China	30.0572	107.874	6.0	9.0	27.0	57.0	75.0	110.0	132.0	...	591.0	591
62	China	26.0789	117.9874	1.0	5.0	10.0	18.0	35.0	59.0	80.0	...	548.0	549
63	China	35.7518	104.2861	0.0	2.0	2.0	4.0	7.0	14.0	19.0	...	187.0	187
64	China	23.3417	113.4244	26.0	32.0	53.0	78.0	111.0	151.0	207.0	...	2184.0	2187
65	China	23.8298	108.7881	2.0	5.0	23.0	23.0	36.0	46.0	51.0	...	267.0	267
66	China	26.8154	106.8748	1.0	3.0	3.0	4.0	5.0	7.0	9.0	...	147.0	147
67	China	19.1959	109.7453	4.0	5.0	8.0	19.0	22.0	33.0	40.0	...	171.0	171
68	China	39.549	116.1306	1.0	1.0	2.0	8.0	13.0	18.0	33.0	...	1317.0	1317
69	China	47.862	127.7615	0.0	2.0	4.0	9.0	15.0	21.0	33.0	...	1610.0	1610
70	China	37.8957	114.9042	5.0	5.0	9.0	32.0	83.0	128.0	168.0	...	1304.0	1304
71	China	22.3	114.2	0.0	2.0	2.0	5.0	8.0	8.0	8.0	...	10848.0	10868
72	China	30.9756	112.2707	444.0	444.0	549.0	761.0	1058.0	1423.0	3554.0	...	68151.0	68151
73	China	27.6104	111.7088	4.0	9.0	24.0	43.0	69.0	100.0	143.0	...	1035.0	1036
74	China	44.0935	113.9448	0.0	0.0	1.0	7.0	7.0	11.0	15.0	...	367.0	367
75	China	32.9711	119.455	1.0	5.0	9.0	18.0	33.0	47.0	70.0	...	703.0	703
76	China	27.614	115.7221	2.0	7.0	18.0	18.0	36.0	72.0	109.0	...	935.0	935
77	China	43.6661	126.1923	0.0	1.0	3.0	4.0	4.0	6.0	8.0	...	573.0	573
78	China	41.2956	122.6085	2.0	3.0	4.0	17.0	21.0	27.0	34.0	...	406.0	406
79	China	22.1667	113.55	1.0	2.0	2.0	2.0	5.0	6.0	7.0	...	48.0	48
80	China	37.2692	106.1655	1.0	1.0	2.0	3.0	4.0	7.0	11.0	...	75.0	75
81	China	35.7452	95.9956	0.0	0.0	0.0	1.0	1.0	6.0	6.0	...	18.0	18
82	China	35.1917	108.8701	0.0	3.0	5.0	15.0	22.0	35.0	46.0	...	547.0	547
83	China	36.3427	118.1498	2.0	6.0	15.0	27.0	46.0	75.0	95.0	...	867.0	867
84	China	31.202	121.4491	9.0	16.0	20.0	33.0	40.0	53.0	66.0	...	1781.0	1783

	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020	1/26/2020	1/27/2020	1/28/2020	...	2/20/2021	2/21/20
85	China	37.5777	112.2922	1.0	1.0	1.0	6.0	9.0	13.0	27.0	...	239.0	239
86	China	30.6171	102.7103	5.0	8.0	15.0	28.0	44.0	69.0	90.0	...	885.0	887
87	China	39.3054	117.323	4.0	4.0	8.0	10.0	14.0	23.0	24.0	...	351.0	352
88	China	31.6927	88.0924	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1
89	China	41.1129	85.2401	0.0	2.0	2.0	3.0	4.0	5.0	10.0	...	980.0	980
90	China	24.974	101.487	1.0	2.0	5.0	11.0	16.0	26.0	44.0	...	231.0	231
91	China	29.1832	120.0934	10.0	27.0	43.0	62.0	104.0	128.0	173.0	...	1320.0	1320

33 rows × 408 columns

In [8]:
```
conf_US = L_conf[250]             #US
recov_US= L_recov[250]
deaths_US = L_deaths[250]

conf_Spain = L_conf[234]          #Spain
recov_Spain= L_recov[234]
deaths_Spain = L_deaths[234]


conf_Italy= L_conf[153]             #Italy
recov_Italy= L_recov[153]
deaths_Italy = L_deaths[153]


conf_Germany = L_conf[134]            #Germany
recov_Germany= L_recov[134]
deaths_Germany = L_deaths[134]

conf_China = L_conf[72]           #China
recov_China= L_recov[72]
deaths_China = L_deaths[72]


days = range(len(conf_US)) #Total number of columns
```

Total Confirm case

In [9]:
```
for ele in range(0, len(conf_US[5:])):
    conf_US_total=0.0
    conf_US_total = (conf_US_total + float(conf_US[5:][ele]))

for ele in range(0, len(conf_Spain[5:])):
    conf_Spain_total=0.0
    conf_Spain_total = (conf_Spain_total + float(conf_Spain[5:][ele]))

for ele in range(0, len(conf_Italy[5:])):
    conf_Italy_total=0.0
    conf_Italy_total = (conf_Italy_total + float(conf_Italy[5:][ele]))

for ele in range(0, len(conf_Germany[5:])):
    conf_Germany_total=0.0
    conf_Germany_total = (conf_Germany_total + float(conf_Germany[5:][ele]))

for ele in range(0, len(conf_China[5:])):
    conf_China_total=0.0
    conf_China_total = (conf_China_total + float(conf_China[5:][ele]))
```

Total death

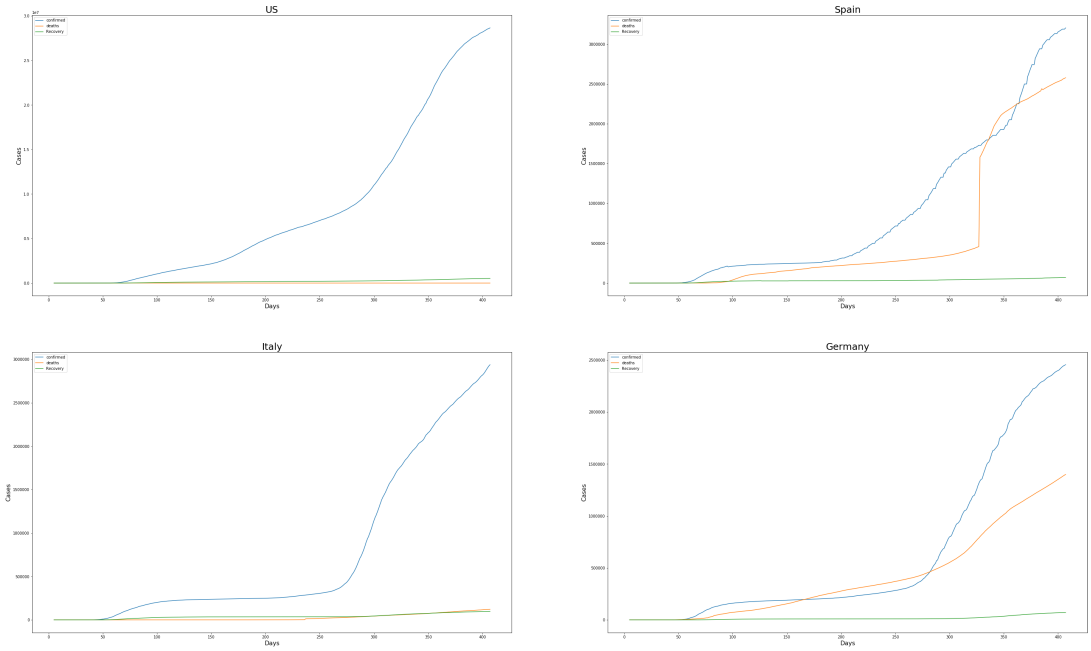
In [10]:
```
for ele in range(0, len(deaths_US[5:])):
    deaths_US_total=0.0
    deaths_US_total = (deaths_US_total + float(deaths_US[5:][ele]))

for ele in range(0, len(deaths_Spain[5:])):
    deaths_Spain_total=0.0
    deaths_Spain_total = (deaths_Spain_total + float(deaths_Spain[5:][ele]))

for ele in range(0, len(deaths_Italy[5:])):
    deaths_Italy_total=0.0
    deaths_Italy_total = (deaths_Italy_total + float(deaths_Italy[5:][ele]))

for ele in range(0, len(deaths_Germany[5:])):
    deaths_Germany_total=0.0
    deaths_Germany_total = (deaths_Germany_total + float(deaths_Germany[5:][ele]))

for ele in range(0, len(deaths_China[5:])):
    deaths_China_total=0.0
    deaths_China_total = (deaths_China_total + float(deaths_China[5:][ele]))
```

Total recover

In [11]:
```
for ele in range(0, len(recov_US[5:])):
    recov_US_total=0.0
    recov_US_total = (recov_US_total + float(recov_US[5:][ele]))

for ele in range(0, len(recov_Spain[5:])):
    recov_Spain_total=0.0
    recov_Spain_total = (recov_Spain_total + float(recov_Spain[5:][ele]))

for ele in range(0, len(recov_Italy[5:])):
    recov_Italy_total=0.0
    recov_Italy_total = (recov_Italy_total + float(recov_Italy[5:][ele]))

for ele in range(0, len(recov_Germany[5:])):
    recov_Germany_total=0.0
    recov_Germany_total = (recov_Germany_total + float(recov_Germany[5:][ele]))

for ele in range(0, len(recov_China[5:])):
    recov_China_total=0.0
    recov_China_total = (recov_China_total + float(recov_China[5:][ele]))
```

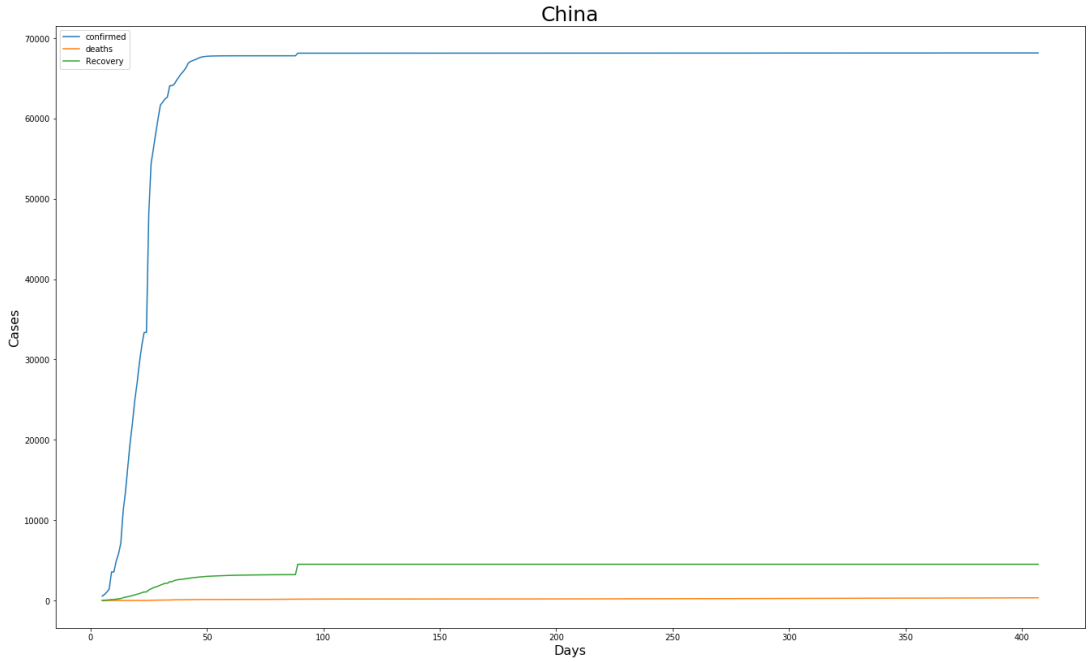
Plot for confirm death and recovery for US,Spain,Italy,Germany

In [12]:
```
plt.figure(figsize = (50,30))

plt.subplot(221)
plt.plot(days[5:], conf_US[5:])
plt.plot(days[5:], recov_US[5:])
plt.plot(days[5:], deaths_US[5:])
plt.legend(['confirmed','deaths', 'Recovery'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('US' , size = 25)
plt.savefig('Cases vs Days_US.png')

plt.subplot(222)
plt.plot(days[5:], conf_Spain[5:])
plt.plot(days[5:], recov_Spain[5:])
plt.plot(days[5:], deaths_Spain[5:])
plt.legend(['confirmed','deaths', 'Recovery'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Spain' , size = 25)
plt.savefig('Cases vs Days_Spain.png')

plt.subplot(223)
plt.plot(days[5:], conf_Italy[5:])
plt.plot(days[5:], recov_Italy[5:])
plt.plot(days[5:], deaths_Italy[5:])
plt.legend(['confirmed','deaths', 'Recovery'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Italy' , size = 25)
plt.savefig('Cases vs Days_Italy.png')

plt.subplot(224)
plt.plot(days[5:], conf_Germany[5:])
plt.plot(days[5:], recov_Germany[5:])
plt.plot(days[5:], deaths_Germany[5:])
plt.legend(['confirmed','deaths', 'Recovery'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Germany' , size = 25)
plt.savefig('Cases vs Days_Germany.png')
```



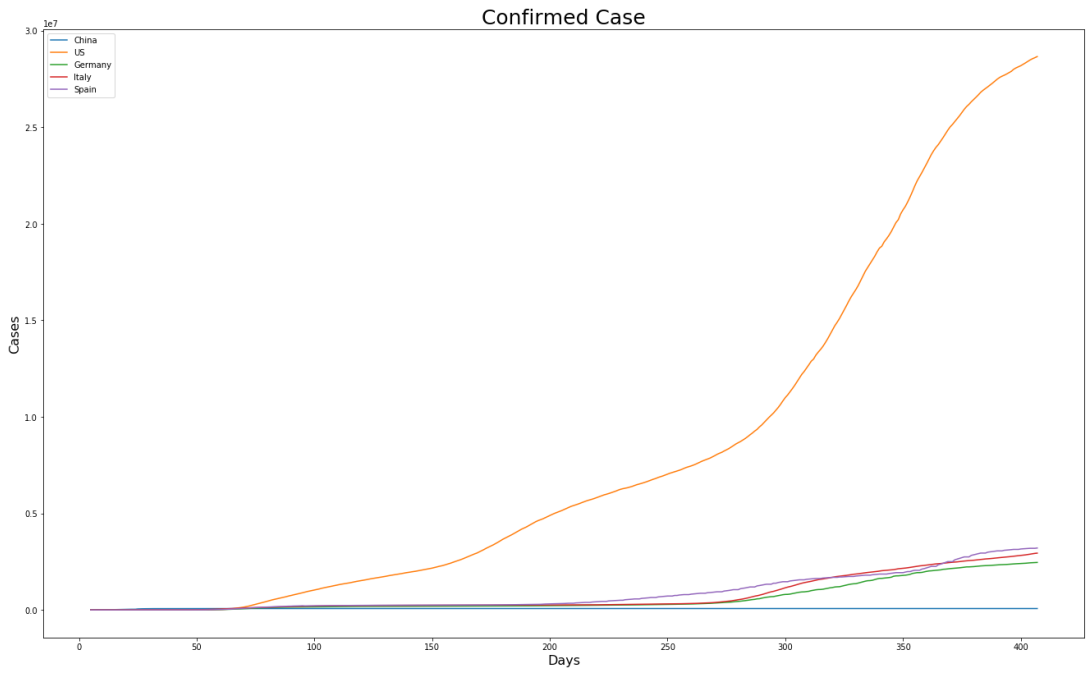
Plot for confirm ,death and recovery for China

In [23]:
```
plt.figure(figsize = (50,30))

plt.subplot(221)
plt.plot(days[5:], conf_China[5:])
plt.plot(days[5:], recov_China[5:])
plt.plot(days[5:], deaths_China[5:])
plt.legend(['confirmed','deaths', 'Recovery'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('China' , size = 25)
plt.savefig('Cases vs Days_china.png')
```

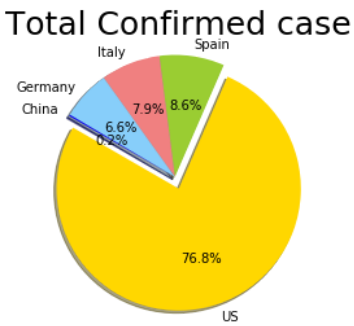


Comparing Confirmed case with China

In [14]:
```
plt.figure(figsize = (50,30))
plt.subplot(224)
plt.plot(days[5:], conf_China[5:])
plt.plot(days[5:], conf_US[5:])
plt.plot(days[5:], conf_Germany[5:])
plt.plot(days[5:], conf_Italy[5:])
plt.plot(days[5:], conf_Spain[5:])
plt.legend(['China','US', 'Germany', 'Italy', 'Spain'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Confirmed Case' , size = 25)
plt.savefig('Confirmed All .png')
```



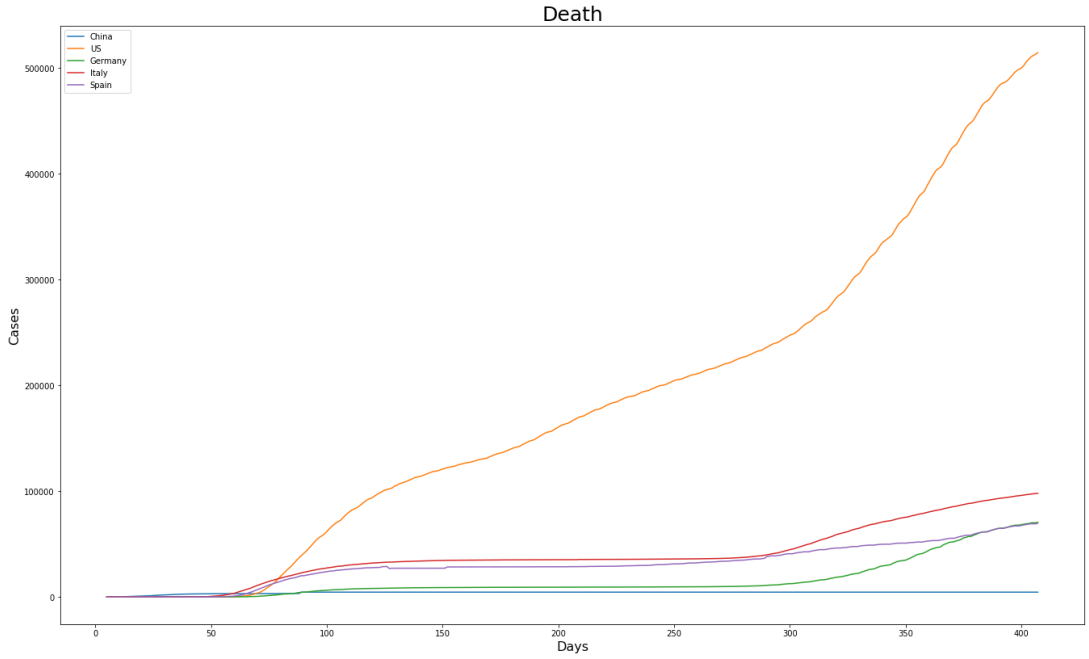
In [15]:
```
# Total confirmed
labels = 'US', 'Spain','Italy', 'Germany', 'China'
sizes = [conf_US_total, conf_Spain_total, conf_Italy_total, conf_Germany_total,conf_China_total]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue','blue']
explode = (0.1, 0, 0, 0,0)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=150)
plt.title('Total Confirmed case' , size = 25)
plt.axis('equal')
plt.show()
plt.savefig('Confirmed all pi.png')
```




```
<Figure size 432x288 with 0 Axes>
```

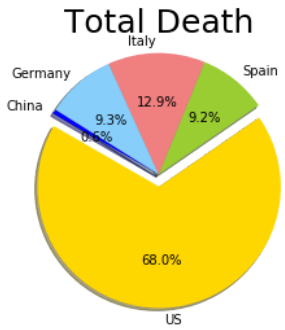
Comparing Death case with China

In [16]:
```
plt.figure(figsize = (50,30))
plt.subplot(224)
plt.plot(days[5:], deaths_China[5:])
plt.plot(days[5:], deaths_US[5:])
plt.plot(days[5:], deaths_Germany[5:])
plt.plot(days[5:], deaths_Italy[5:])
plt.plot(days[5:], deaths_Spain[5:])
plt.legend(['China','US', 'Germany', 'Italy', 'Spain'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Death' , size = 25)
plt.savefig('Death all.png')
```

In [ ]:
```
```

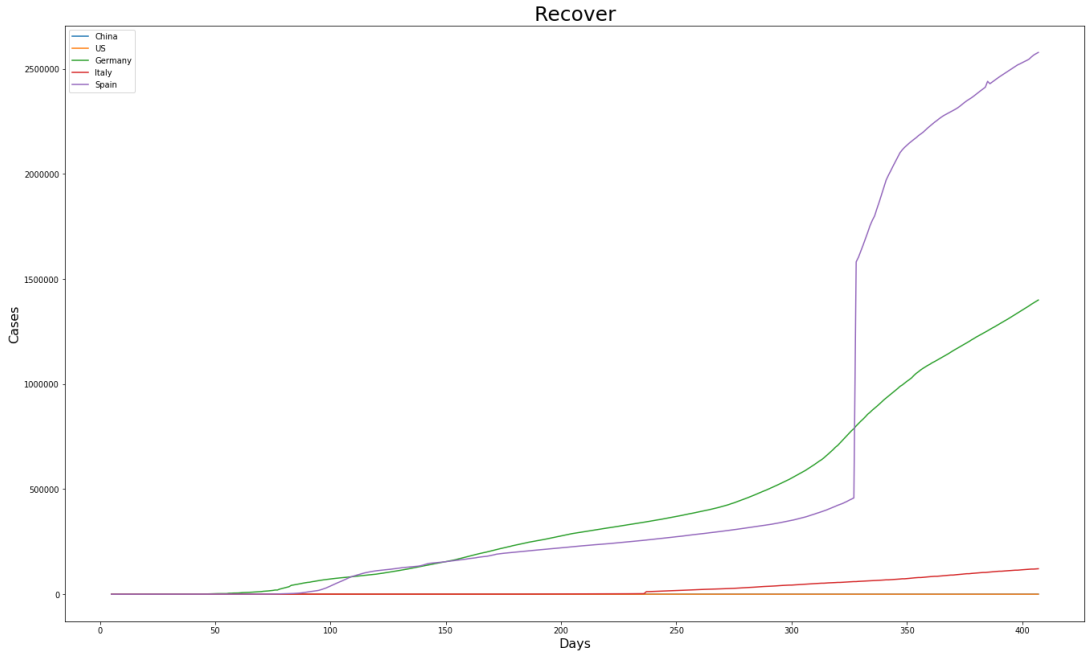
In [17]:
```
# Total deaths
labels = 'US', 'Spain','Italy', 'Germany', 'China'
sizes = [deaths_US_total, deaths_Spain_total, deaths_Italy_total, deaths_Germany_total,deaths_China_total]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue','blue']
explode = (0.1, 0, 0, 0,0)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=150)
plt.title('Total Death' , size = 25)

plt.axis('equal')
plt.show()
plt.savefig('Death all pi.png')
```



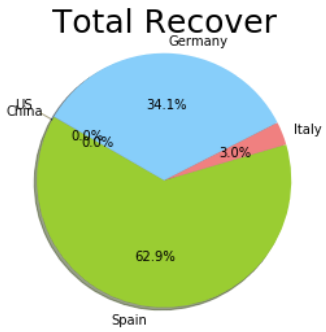

```
<Figure size 432x288 with 0 Axes>
```

Comparing Recover case with China

In [18]:
```
plt.figure(figsize = (50,30))
plt.subplot(224)
plt.plot(days[5:], recov_China[5:])
plt.plot(days[5:], recov_US[5:])
plt.plot(days[5:], recov_Germany[5:])
plt.plot(days[5:], recov_Italy[5:])
plt.plot(days[5:], recov_Spain[5:])
plt.legend(['China','US', 'Germany', 'Italy', 'Spain'])
plt.xlabel('Days', size = 16)
plt.ylabel('Cases' , size = 16)
plt.title('Recover' , size = 25)
plt.savefig('Recover all.png')
```



In [19]:
```
# Total cRecover
labels = 'US', 'Spain','Italy', 'Germany', 'China'
sizes = [recov_US_total, recov_Spain_total, recov_Italy_total, recov_Germany_total,recov_China_total]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue','blue']
explode = (0.1, 0, 0, 0,0)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=150)
plt.title('Total Recover' , size = 25)

plt.axis('equal')
plt.show()
plt.savefig('Recover pi.png')
```




```
<Figure size 432x288 with 0 Axes>
```