```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
percentage=np.array([0.54,0.24,0.05,0.17])
waste=np.array([142566,141064,101066.27,119140.9,135143.05,152076.7])
population=np.array([1295.6,1310.2,1324.5,1338.7,1352.6,1366.4])
gdp=np.array([1.86,2.04,2.1,2.29,2.65,2.7])*(10**5)
year=13
household=[]
commercial=[]
agricultural=[]
ewaste=[]
percapita_household=[]
percapita_commercial=[]
percapita_agricultural=[]
percapita_ewaste=[]
waste13=percentage*waste[0]
waste14=percentage*waste[1]
waste15=percentage*waste[2]
waste16=percentage*waste[3]
waste17=percentage*waste[4]
waste18=percentage*waste[5]

for i in range(len(waste)):
  print(year)
  print("household:",waste[i]*percentage[0])
  household.append(waste[i]*percentage[0])
  print("commercial:",waste[i]*percentage[1])
  commercial.append(waste[i]*percentage[1])
  print("Agricultural:",waste[i]*percentage[2])
  agricultural.append(waste[i]*percentage[2])
  print("E-waste:",waste[i]*percentage[3])
  ewaste.append(waste[i]*percentage[3])
  year=year+1

household=np.asarray(household)/1000
commercial=np.asarray(commercial)/1000
agricultural=np.asarray(agricultural)/1000
ewaste=np.asarray(ewaste)/1000

for i in range(len(waste)):
  percapita_household.append(household[i]*1000/population[i])
  percapita_commercial.append(commercial[i]*1000/population[i])
  percapita_agricultural.append(agricultural[i]*1000/population[i])
  percapita_ewaste.append(ewaste[i]*1000/population[i])

percapita_household=np.asarray(percapita_household)
percapita_commercial=np.asarray(percapita_commercial)
percapita_agricultural=np.asarray(percapita_agricultural)
percapita_ewaste=np.asarray(percapita_ewaste)

def A(x):
  print("Minimum:",np.min(x))
```

```
  print("Maximum:",np.max(x))
  print("Standard deviation:",np.std(x))
  print("Mean:",np.mean(x))
  print("standard error: ",np.std(x)/np.sqrt(len(x)))
  return
```

```
13
household: 76985.64
commercial: 34215.84
Agricultural: 7128.3
E-waste: 24236.22
14
household: 76174.56000000001
commercial: 33855.36
Agricultural: 7053.200000000001
E-waste: 23980.88
15
household: 54575.785800000005
commercial: 24255.9048
Agricultural: 5053.3135
E-waste: 17181.265900000002
16
household: 64336.086
commercial: 28593.816
Agricultural: 5957.045
E-waste: 20253.953
17
household: 72977.247
commercial: 32434.331999999995
Agricultural: 6757.1525
E-waste: 22974.3185
18
household: 82121.418
commercial: 36498.408
Agricultural: 7603.835000000001
E-waste: 25853.039000000004
```

```
A(household)
print("---------------------------")
A(percapita_household)
```

```
Minimum: 54.575785800000006
Maximum: 82.121418
Standard deviation: 9.16636538181239
Mean: 71.19512280000001
standard error:  3.742152996892044
---------------------------
Minimum: 41.20482129105323
Maximum: 60.100569379391104
Standard deviation: 6.845108833020406
Mean: 53.479635280044704
standard error:  2.7945039791196695
```

```
A(commercial)
print("---------------------------")
A(percapita_commercial)
```

```
Minimum: 24.2559048
Maximum: 36.498408000000005
Standard deviation: 4.073940169694396
Mean: 31.642276800000005
standard error:  1.6631791097297972
---------------------------
Minimum: 18.31325390713477
Maximum: 26.711364168618267
Standard deviation: 3.0422705924535136
Mean: 23.768726791130973
standard error:  1.2420017684976308
```

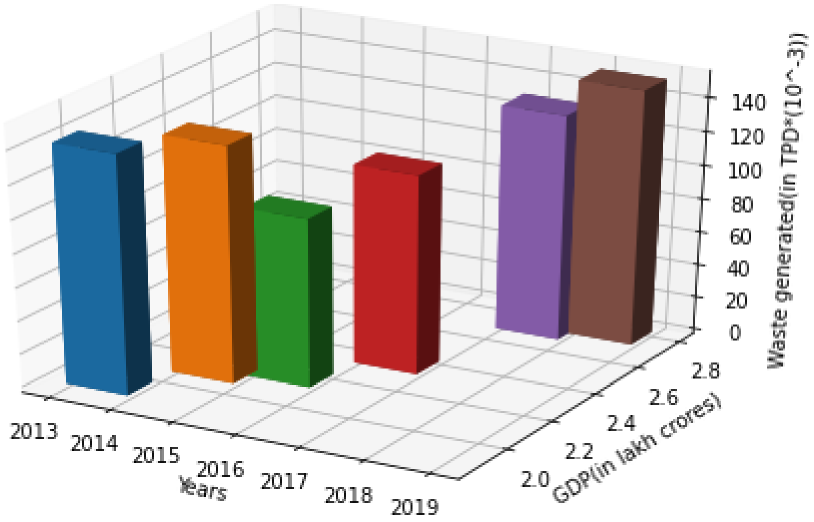
```
A(agricultural)
print("---------------------------")
A(percapita_agricultural)
```

```
Minimum: 5.0533135
Maximum: 7.603835000000001
Standard deviation: 0.8487375353529993
Mean: 6.592141000000001
standard error:  0.3464956478603745
---------------------------
Minimum: 3.8152612306530767
Maximum: 5.564867535128806
Standard deviation: 0.6338063734278155
Mean: 4.951818081485619
standard error:  0.2587503684370065
```

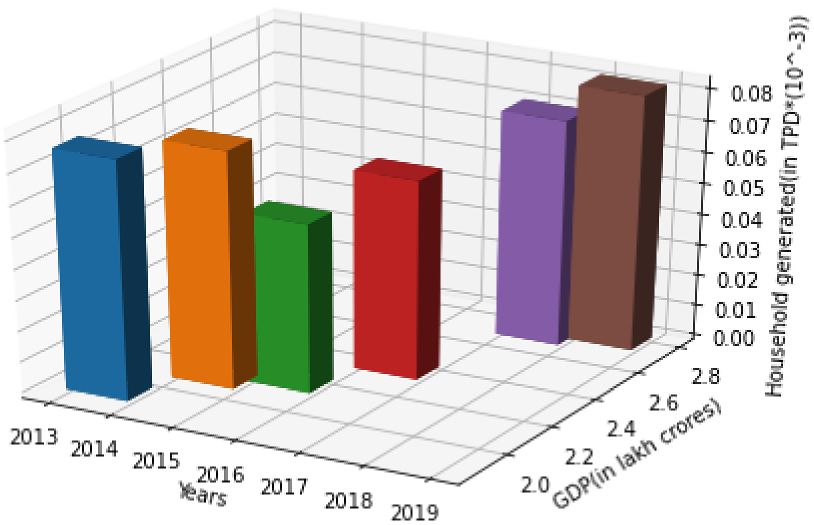
```
A(ewaste)
print("---------------------------")
A(percapita_ewaste)
```

```
Minimum: 17.181265900000003
Maximum: 25.853039000000003
Standard deviation: 2.885707620200196
Mean: 22.413279400000004
standard error:  1.1780852027252726
---------------------------
Minimum: 12.971888184220463
Maximum: 18.920549619437942
Standard deviation: 2.1549416696545722
Mean: 16.83618147705111
standard error:  0.8797512526858218
```

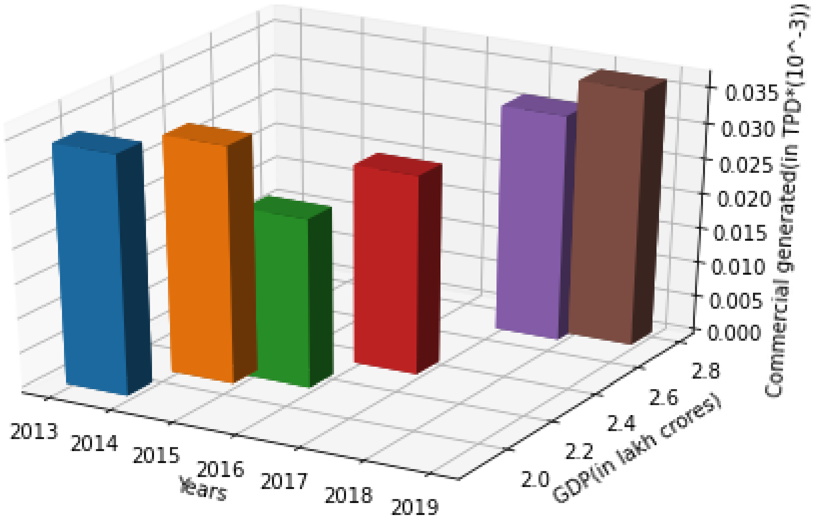
```
years=np.array([2014,2015,2016,2017,2018,2019])
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,gdp[i]/10**5, 0, 1, 0.1,waste[i]/1000)
ax.set_xlabel("Years")
ax.set_ylabel("GDP(in lakh crores)")
ax.set_zlabel("Waste generated(in TPD*(10^-3))")
plt.savefig("3dplot1")
plt.show()
```



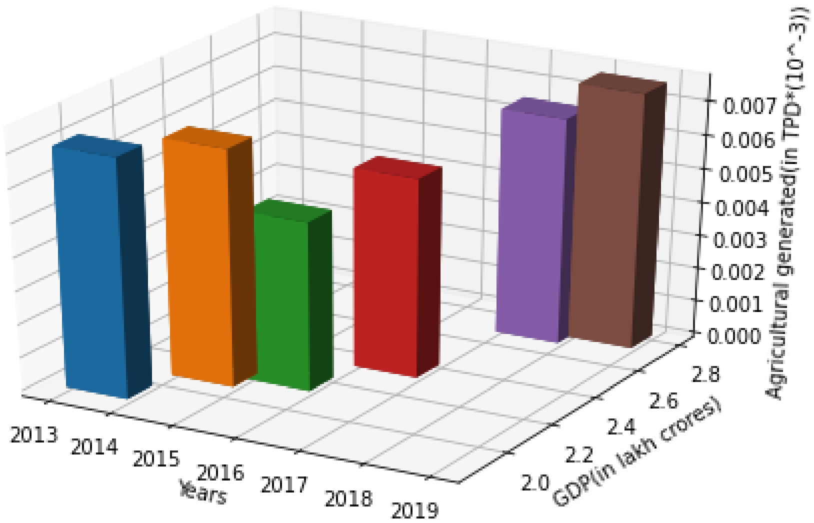

```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,gdp[i]/10**5, 0, 1, 0.1,household[i]/1000)
ax.set_xlabel("Years")
ax.set_ylabel("GDP(in lakh crores)")
ax.set_zlabel("Household generated(in TPD*(10^-3))")
plt.savefig("3dplot2")
plt.show()
```



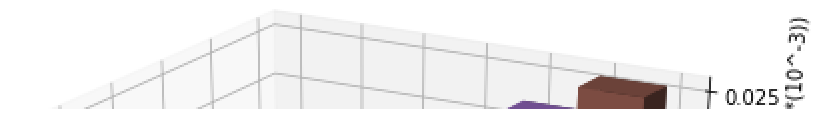

```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,gdp[i]/10**5, 0, 1, 0.1,commercial[i]/1000)
ax.set_xlabel("Years")
ax.set_ylabel("GDP(in lakh crores)")
ax.set_zlabel("Commercial generated(in TPD*(10^-3))")


plt.savefig("3dplot3")
plt.show()
```

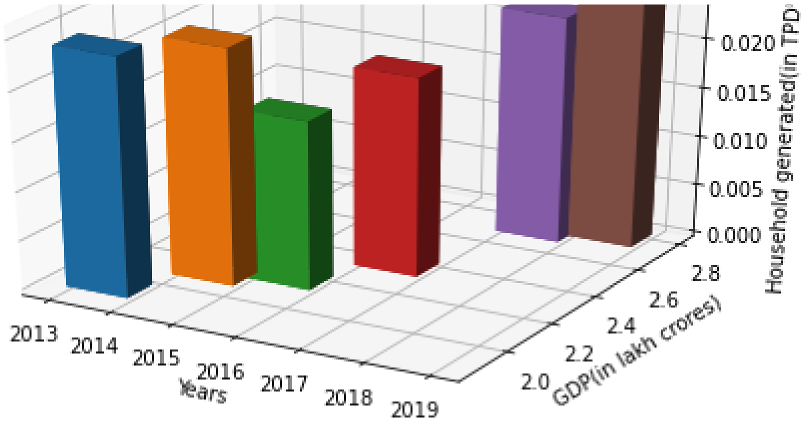



```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,gdp[i]/10**5, 0, 1, 0.1,agricultural[i]/1000)
ax.set_xlabel("Years")
ax.set_ylabel("GDP(in lakh crores)")
ax.set_zlabel("Agricultural generated(in TPD*(10^-3))")
plt.savefig("3dplot4")
plt.show()
```

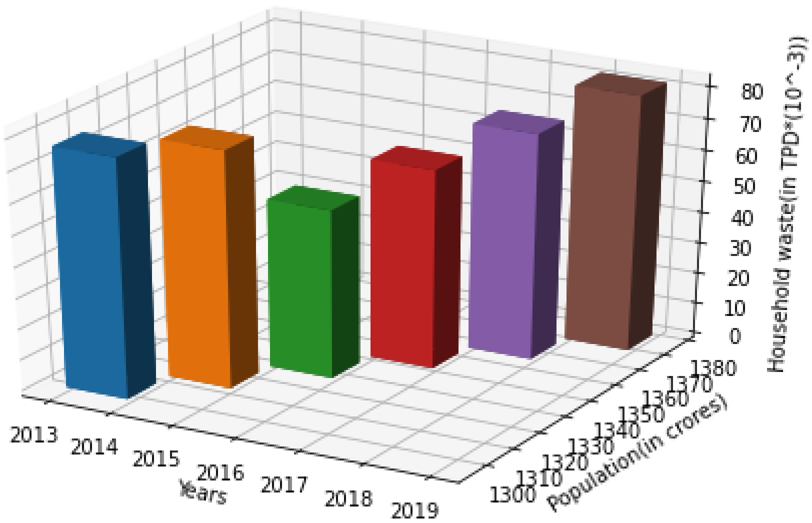



```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,gdp[i]/10**5, 0, 1, 0.1,ewaste[i]/1000)
ax.set_xlabel("Years")
ax.set_ylabel("GDP(in lakh crores)")
ax.set_zlabel("Household generated(in TPD*(10^-3))")
plt.savefig("3dplot5")
plt.show()
```

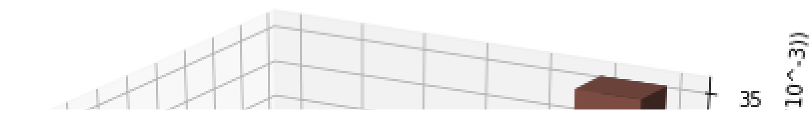


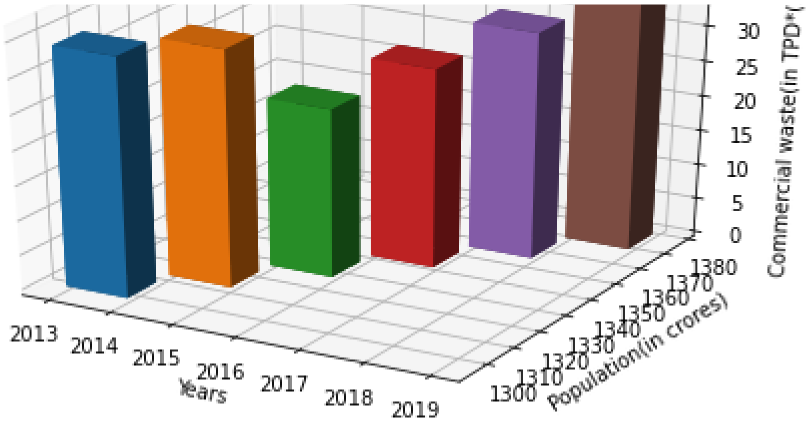



```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,population[i], 0, 1, 10,household[i])
ax.set_xlabel("Years")
ax.set_ylabel("Population(in crores)")
ax.set_zlabel("Household waste(in TPD*(10^-3))")
plt.savefig("3dplot6")
plt.show()
```

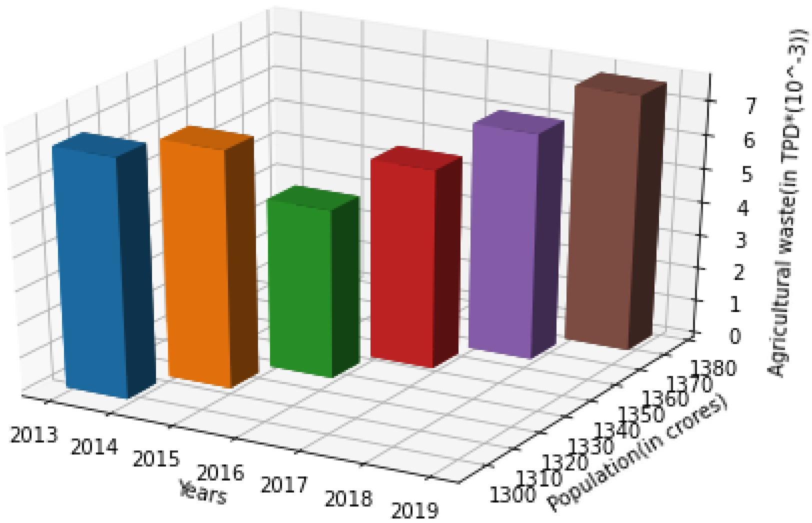



```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,population[i], 0, 1, 10,commercial[i])
ax.set_xlabel("Years")
ax.set_ylabel("Population(in crores)")
ax.set_zlabel("Commercial waste(in TPD*(10^-3))")
plt.savefig("3dplot7")
plt.show()
```

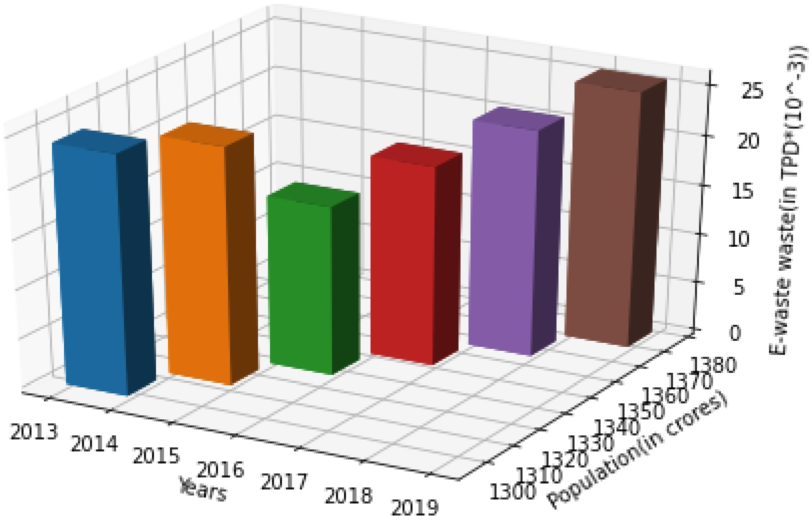




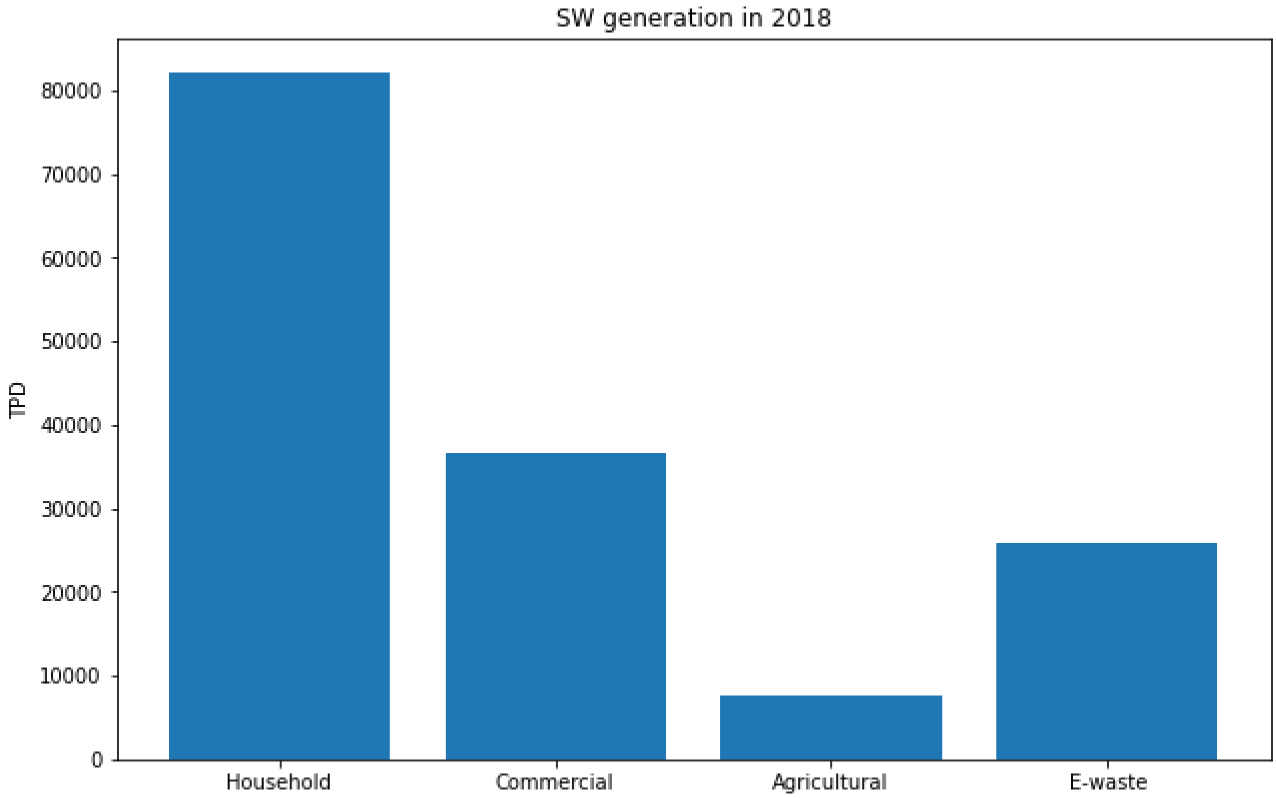

```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
for i in range(6):
  ax.bar3d(years[i]-1,population[i], 0, 1, 10,agricultural[i])
ax.set_xlabel("Years")
ax.set_ylabel("Population(in crores)")
ax.set_zlabel("Agricultural waste(in TPD*(10^-3))")
plt.savefig("3dplot8")
plt.show()
```



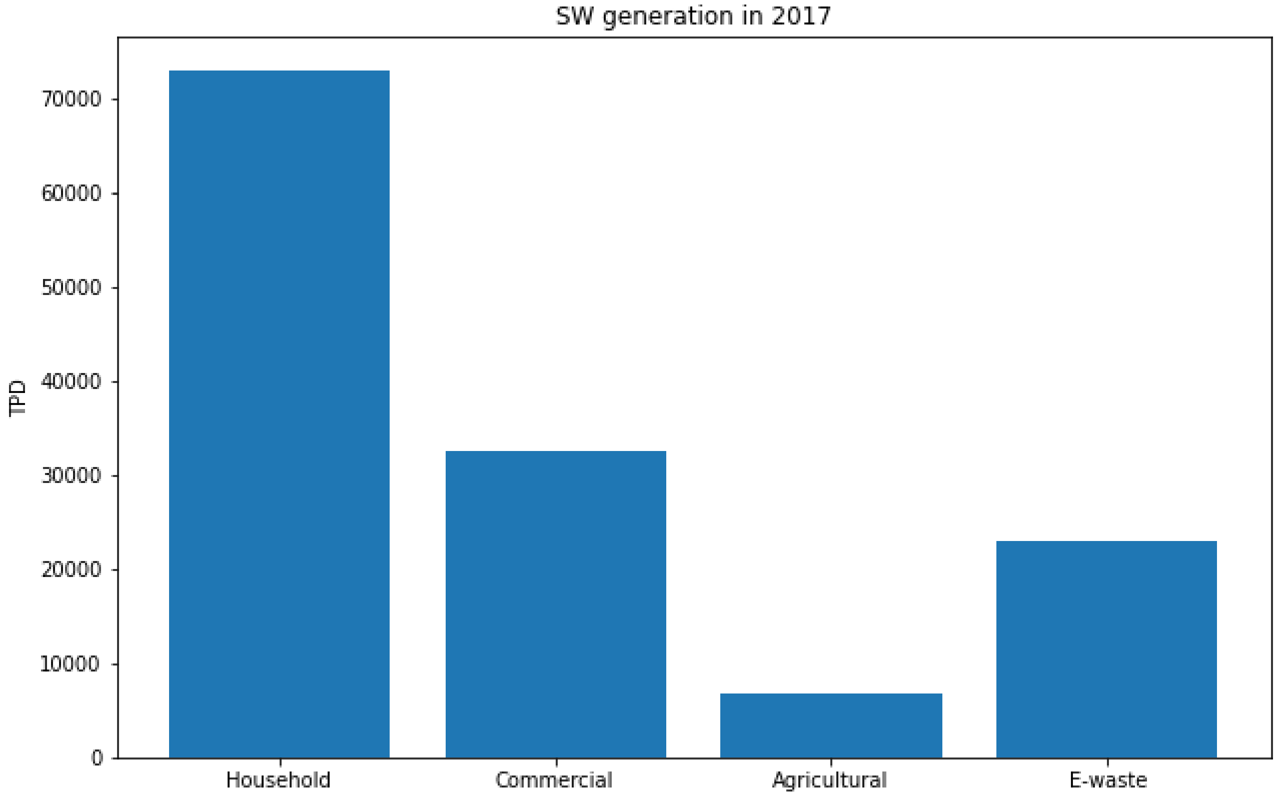

```
fig = plt.figure(figsize=(8,5))
ax = plt.axes(projection="3d")
num_bars = 7
for i in range(6):
  ax.bar3d(years[i]-1,population[i], 0, 1, 10,ewaste[i])
ax.set_xlabel("Years")
ax.set_ylabel("Population(in crores)")
ax.set_zlabel("E-waste waste(in TPD*(10^-3))")


plt.savefig("3dplot9")
plt.show()
```




```
types=['Household','Commercial','Agricultural','E-waste']
fig = plt.figure(figsize=(8,5))
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2018')
ax.bar(types,waste18)
plt.savefig('plot1.png')
plt.show()
```



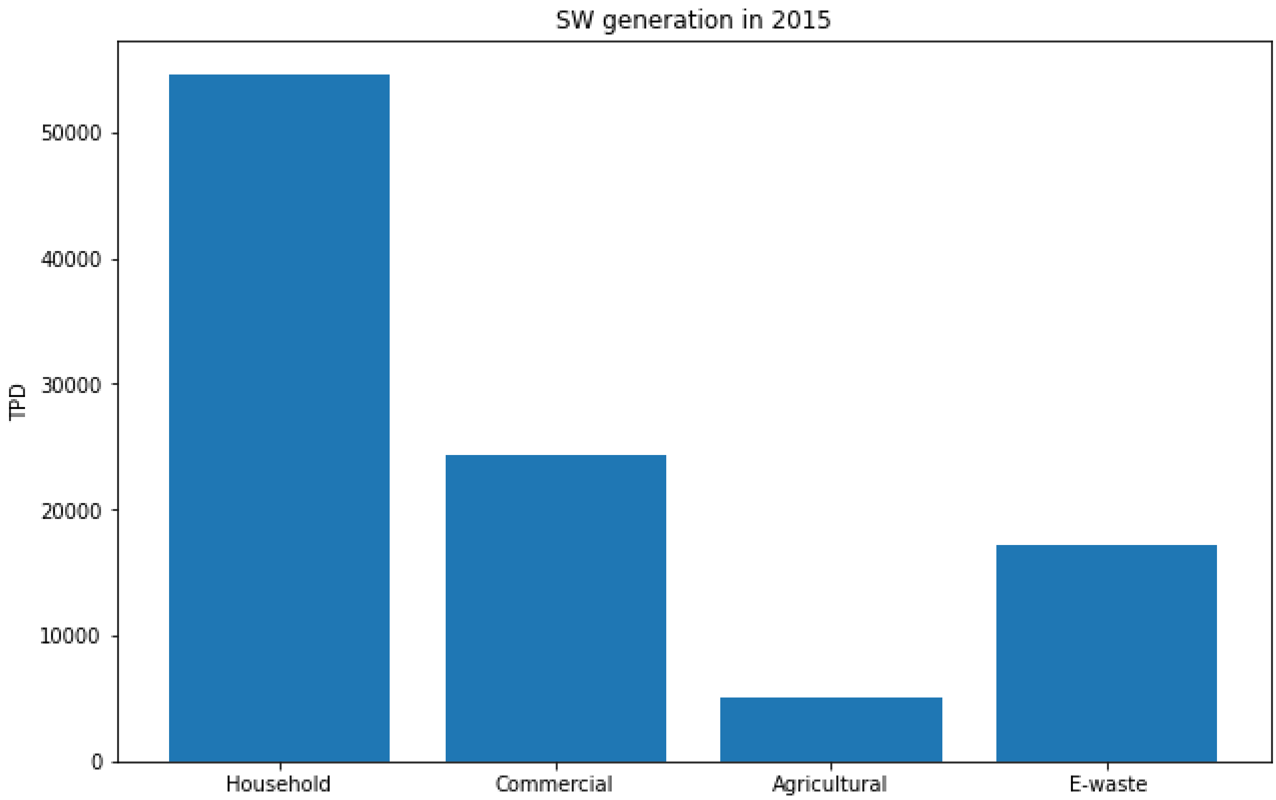

```
fig = plt.figure(figsize=(8,5))
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2017')
ax.bar(types,waste17)
plt.savefig('plot2.png')
plt.show()
```



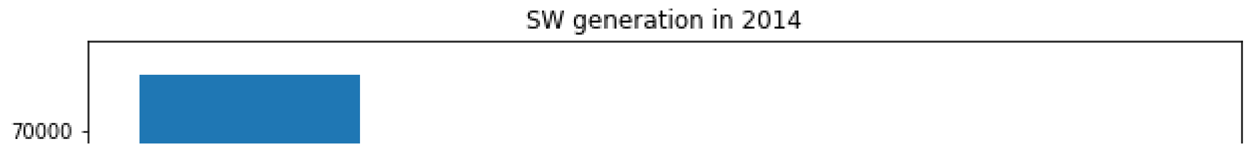

```
fig = plt.figure(figsize=(8,5))
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2016')
ax.bar(types,waste16)
plt.savefig('plot3.png')
plt.show()
```


```
fig = plt.figure(figsize=(8,5))
```

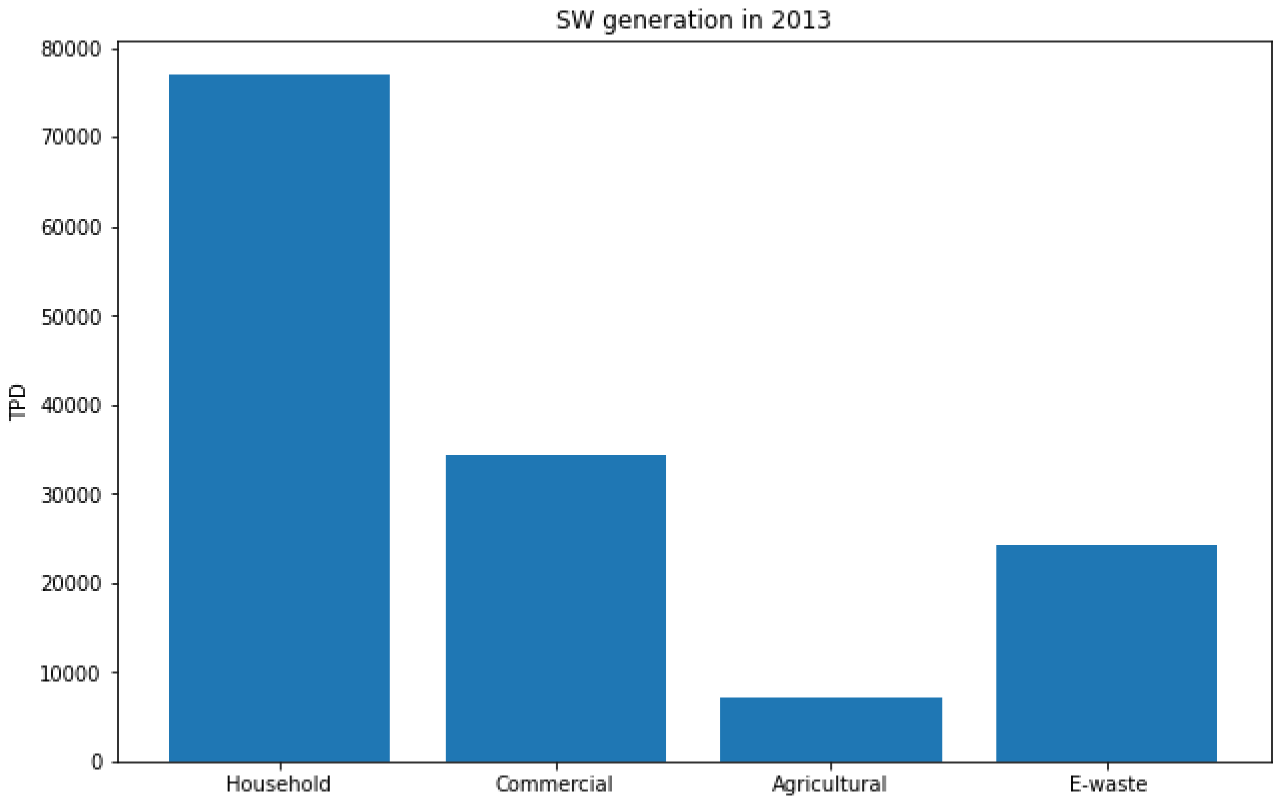
```
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2015')
ax.bar(types,waste15)
plt.savefig('plot4.png')
plt.show()
```




```
fig = plt.figure(figsize=(8,5))
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2014')
ax.bar(types,waste14)
plt.savefig('plot5.png')
plt.show()
```

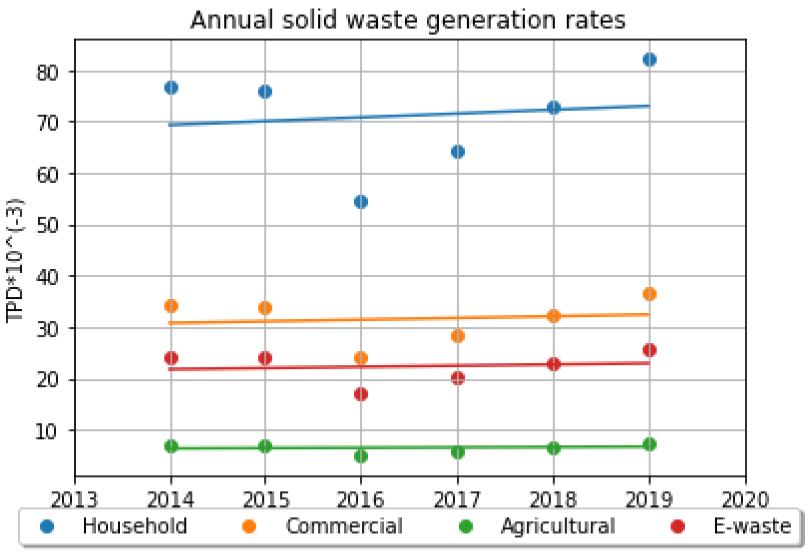



```
fig = plt.figure(figsize=(8,5))
ax = fig.add_axes([0,0,1,1])
ax.set_ylabel('TPD')
ax.set_title('SW generation in 2013')
ax.bar(types,waste13)
plt.savefig('plot6.png')
plt.show()
```




```
years=np.array([2014,2015,2016,2017,2018,2019])
ax = plt.subplot(111)
plt.scatter(years,household,label='Household')
m1,b1=np.polyfit(years,household,1)
plt.plot(years, m1*years+b1)
plt.scatter(years,commercial,label='Commercial')
m2,b2=np.polyfit(years,commercial,1)
plt.plot(years, m2*years+b2)
plt.scatter(years,agricultural,label='Agricultural')
m3,b3=np.polyfit(years,agricultural,1)
plt.plot(years, m3*years+b3)
plt.scatter(years,ewaste,label='E-waste')
m4,b4=np.polyfit(years,ewaste,1)

plt.plot(years, m4*years+b4)
plt.grid(True)
plt.title("Annual solid waste generation rates")
plt.xlabel("Years")
```

```
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
ax.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05),
          fancybox=True, shadow=True, ncol=5)
plt.savefig("fig1")
plt.show()
```




```
print("m1,b1 :",m1,b1)
print("m2,b2 :",m2,b2)
print("m3,b3 :",m3,b3)
print("m4,b4 :",m4,b4)
```

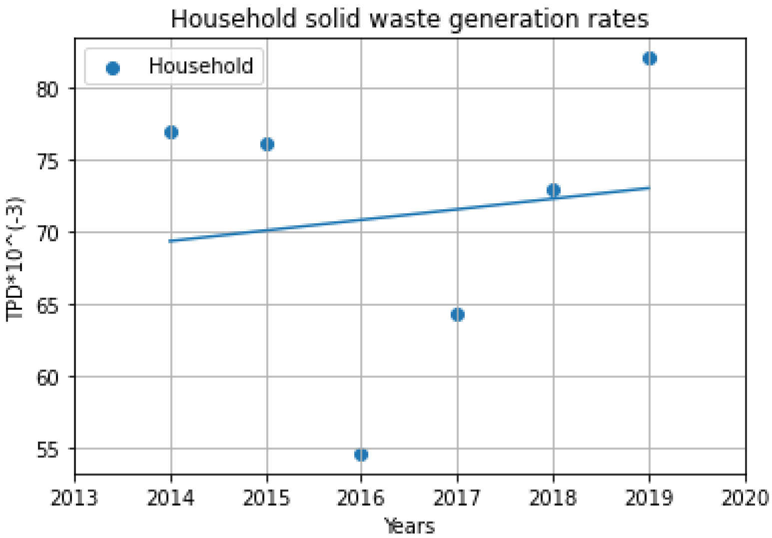
```
m1,b1 : 0.7384928914279942 -1417.9757927645496
m2,b2 : 0.3282190628568851 -630.2114634509086
m3,b3 : 0.06837897142851801 -131.2940548856065
m4,b4 : 0.23248850285696146 -446.3997866110625
```

```
#projection of household,commercial,agricultural,ewaste for the years in x1 array using li
x1=np.array([2022,2023,2024,2025,2026])
y1=m1*x1+b1
y2=m2*x1+b2
y3=m3*x1+b3
y4=m4*x1+b4
print(y1)
print(y2)
print(y3)
print(y4)
```

```
[75.2568337  75.99532659 76.73381949 77.47231238 78.21080527]
[33.44748165 33.77570071 34.10391977 34.43213883 34.7603579 ]
[6.96822534 7.03660431 7.10498329 7.17336226 7.24174123]
[23.69196617 23.92445467 24.15694317 24.38943167 24.62192018]
```

```
# plots of waste with linear regression line
plt.scatter(years,household,label='Household')
m11,b11=np.polyfit(years,household,1)
plt.plot(years, m11*years+b11)
```

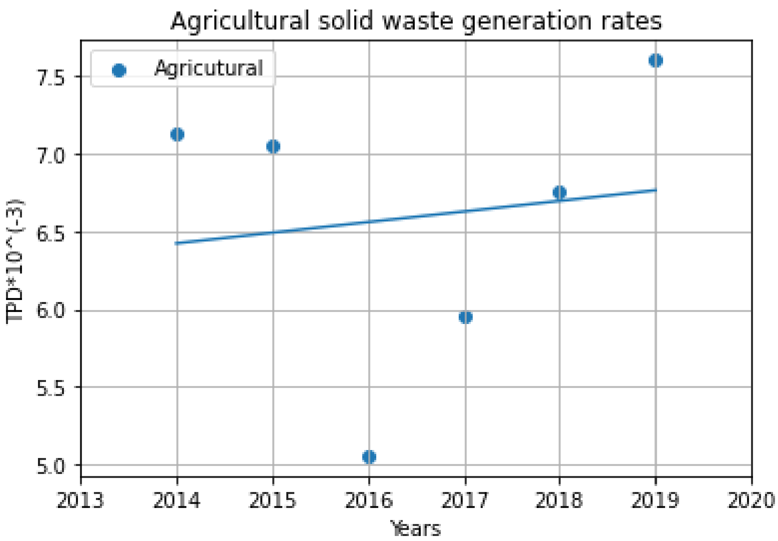
```
plt.grid(True)
plt.title("Household solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig9")
plt.show()
```




```
plt.scatter(years,commercial,label='Commercial')
m21,b21=np.polyfit(years,commercial,1)
plt.plot(years, m21*years+b21)
plt.grid(True)
plt.title("Commercial solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig8")
plt.show()
```

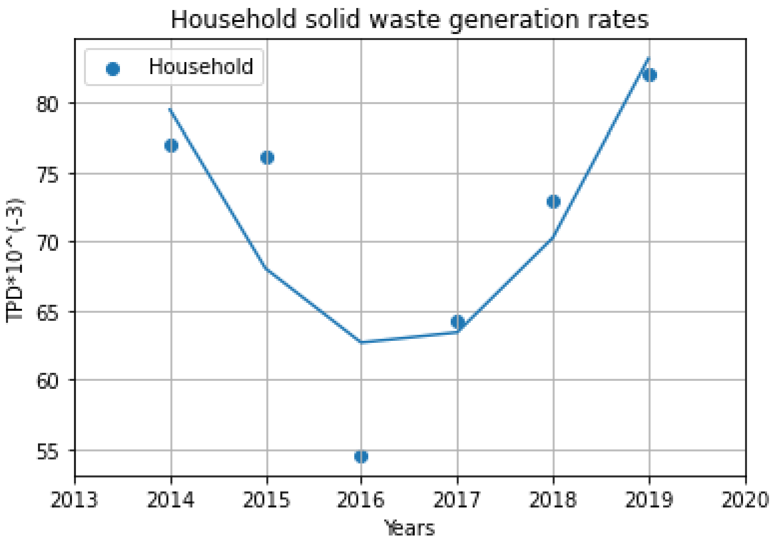

```
plt.scatter(years,agricultural,label='Agricutural')
m31,b31=np.polyfit(years,agricultural,1)
```

```
plt.plot(years, m31*years+b31)
plt.grid(True)
plt.title("Agricultural solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig7")
plt.show()
```

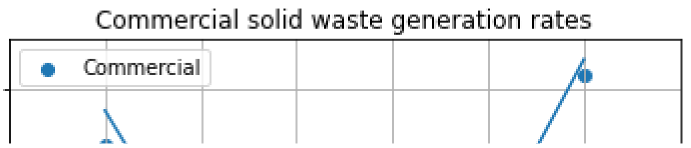


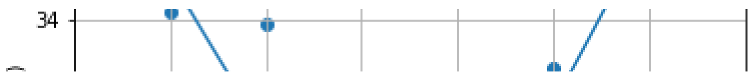

```
plt.scatter(years,ewaste,label='E-waste')
m41,b41=np.polyfit(years,ewaste,1)
plt.plot(years, m41*years+b41)
plt.grid(True)
plt.title("E-waste solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig6")
plt.show()
```

```
#plots with Quadratic regression boundary
plt.scatter(years,household,label='Household')
m1,n1,b1=np.polyfit(years,household,2)
plt.plot(years, m1*years**2+n1*years+b1)
plt.grid(True)
plt.title("Household solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig2")
plt.show()
```

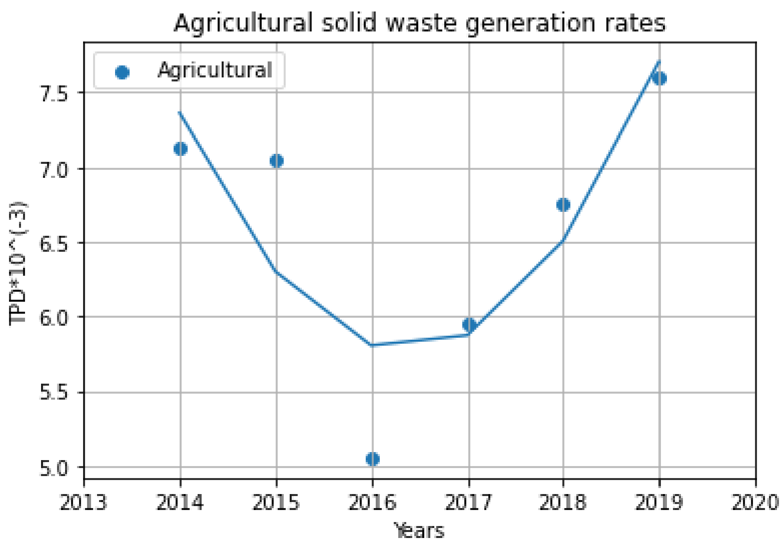



```
plt.scatter(years,commercial,label='Commercial')
m2,n2,b2=np.polyfit(years,commercial,2)
plt.plot(years, m2*years**2+n2*years+b2)
plt.grid(True)
plt.title("Commercial solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig3")
plt.show()
```

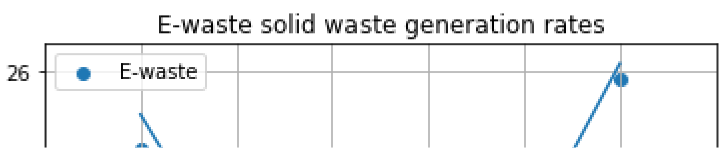


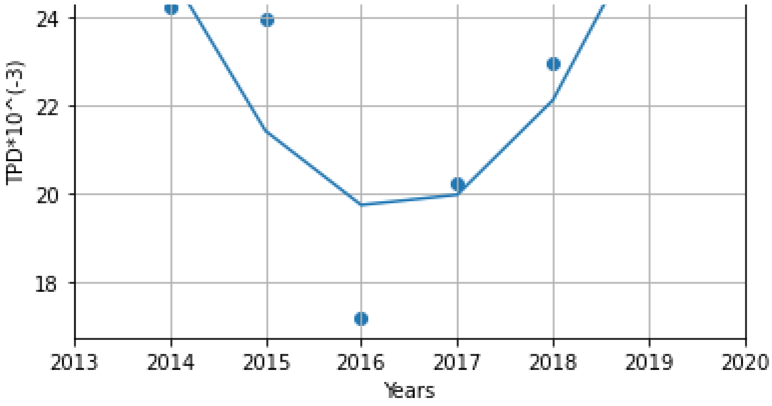



```
plt.scatter(years,agricultural,label='Agricultural')
m3,n3,b3=np.polyfit(years,agricultural,2)
plt.plot(years, m3*years**2+n3*years+b3)
plt.grid(True)
plt.title("Agricultural solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig4")
plt.show()
```




```
plt.scatter(years,ewaste,label='E-waste')
m4,n4,b4=np.polyfit(years,ewaste,2)
plt.plot(years, m4*years**2+n4*years+b4)
plt.grid(True)
plt.title("E-waste solid waste generation rates")
plt.xlabel("Years")
plt.ylabel("TPD*10^(-3)")
plt.xlim(2013,2020)
plt.legend()
plt.savefig("fig5")
plt.show()
```





0s completed at 13:27