Singapore Recycling and Waste Management

Learn how much Singapore is saving energy per years by recycling plastics, paper, glass, ferrous and non-ferrous metal.

> **before going throught my notebook pleas check our my medium article explaining everything in detail.**

In this project, we will clean our data and prepares for data analysis. We will be using Singapore NEA Energy Savings | Kaggle data to analyze the total garbage collection and recycling rate. The material names are different as these data were collected from different resources. We will be added the latest data of 2020 waste-statistics-and-overall-recycling from the website so that we can have the latest statics analysis. We will be finding how much energy we can produce using Greentumble key information.

We will be using **Recycling statistics** to calculate energy saved every year from 2003 to 2020 based on five waste types, plastics, paper, glass, ferrous and non-ferrous metal.

Loading Data

In [1]:
```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go

energy_saved = pd.read_csv("C:/Users/HP/Downloads/waste_energy_stat.csv")
waste_03_17 = pd.read_csv("C:/Users/HP/Downloads/2003_2017_waste.csv")
waste_18_20 = pd.read_csv("C:/Users/HP/Downloads/2018_2020_waste.csv")
```

In [2]:
```
energy_saved
```

Out[2]:

	The table gives the amount of energy saved in kilowatt hour (kWh) and the amount of crude oil (barrels) by recycling 1 metric tonne (1000 kilogram) per waste type	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	1 barrel oil is approximately 159 litres of oil	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	material	Plastic	Glass	Ferrous Metal	Non-Ferrous Metal	Paper
3	energy_saved	5774 Kwh	42 Kwh	642 Kwh	14000 Kwh	4100 kWh
4	crude_oil saved	16 barrels	0.12 barrels	1.8 barrels	40 barrels	11 barrels

In [3]:
```
waste_03_17
```

Out[3]:

	waste_type	waste_disposed_of_tonne	total_waste_recycled_tonne	total_waste_generated_tonne	recycling_rate	year
0	Food	679900	111100.0	791000	0.14	2016
1	Paper/Cardboard	576000	607100.0	1183100	0.51	2016
2	Plastics	762700	59500.0	822200	0.07	2016
3	C&D	9700	1585700.0	1595400	0.99	2016
4	Horticultural waste	111500	209000.0	320500	0.65	2016
...	...	...	...	...	...	...
220	Ash and sludge	214800	28600.0	243400	0.12	2017
221	Plastic	763400	51800.0	815200	0.06	2017
222	Textile/Leather	141200	9600.0	150800	0.06	2017
223	Others (stones, ceramic, rubber, etc.)	319300	7100.0	326400	0.02	2017
224	Total	2980000	4724300.0	7704300	0.61	2017

225 rows × 6 columns

In [4]:
```
waste_18_20
```

Out[4]:

	Waste Type	Total Generated ('000 tonnes)	Total Recycled ('000 tonnes)	Year
0	Construction& Demolition	1624	1618	2018
1	Ferrous Metal	1269	126	2018
2	Paper/Cardboard	1054	586	2018
3	Plastics	949	41	2018
4	Food	763	126	2018
5	Wood	521	428	2018
6	Horticultural	320	227	2018
7	Ash & Sludge	240	25	2018
8	Textile/Leather	220	14	2018
9	Used Slag	181	179	2018
10	Non-Ferrous Metal	171	170	2018
11	Glass	64	12	2018
12	Scrap Tyres	32	29	2018
13	Others (stones, ceramic, rubber, ect)	286	11	2018
14	Overall	7695	4726	2018
15	Construction& Demolition	1440	1434	2019
16	Ferrous Metal	1278	1270	2019
17	Paper/Cardboard	1011	449	2019
18	Plastics	930	37	2019
19	Food	7440	136	2019
20	Wood	438	289	2019
21	Horticultural	400	293	2019
22	Ash & Sludge	252	25	2019
23	Textile/Leather	168	6	2019
24	Used Slag	129	127	2019
25	Non-Ferrous Metal	126	124	2019
26	Glass	75	11	2019
27	Scrap Tyres	33	31	2019
28	Others (stones, ceramic, rubber, ect)	210	15	2019
29	Overall	7234	4247	2019
30	Paper/Cardboard	1144	432	2020
31	Ferrous metal	934	930	2020
32	Plastics	868	36	2020
33	Construction & Demolition	825	822	2020
34	Food	665	126	2020
35	Horticultural	313	249	2020
36	Wood	304	195	2020
37	Ash & sludge	228	16	2020
38	Textile/Leather	137	6	2020
39	Used slag	106	104	2020
40	Non-ferrous metal	75	73	2020
41	Glass	66	7	2020
42	Scrap tyres	23	22	2020
43	Others (stones, ceramics, etc.)	193	21	2020
44	Overall	5880	3040	2020

Cleaning Data

In [5]:

```
clean_waste_18_20 = waste_18_20.rename(
    columns={
        "Waste Type": "waste_type",
        "Total Generated ('000 tonnes)": "total_waste_generated_tonne",
        "Total Recycled ('000 tonnes)": "total_waste_recycled_tonne",
        "Year": "year",
    }
)
```

```
clean_waste_18_20["total_waste_generated_tonne"] = (
    clean_waste_18_20["total_waste_generated_tonne"] * 1000
)
clean_waste_18_20["total_waste_recycled_tonne"] = (
    clean_waste_18_20["total_waste_recycled_tonne"] * 1000
)
```

In [7]:
```
clean_waste_18_20["recycling_rate"] = round(
    clean_waste_18_20["total_waste_recycled_tonne"]
    / clean_waste_18_20["total_waste_generated_tonne"],
    2,
)
clean_waste_18_20.head()
```

Out[7]:

	waste_type	total_waste_generated_tonne	total_waste_recycled_tonne	year	recycling_rate
0	Construction& Demolition	1624000	1618000	2018	1.00
1	Ferrous Metal	1269000	126000	2018	0.10
2	Paper/Cardboard	1054000	586000	2018	0.56
3	Plastics	949000	41000	2018	0.04
4	Food	763000	126000	2018	0.17

In [8]:
```
energy_saved
```

Out[8]:

	The table gives the amount of energy saved in kilowatt hour (kWh) and the amount of crude oil (barrels) by recycling 1 metric tonne (1000 kilogram) per waste type	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	1 barrel oil is approximately 159 litres of oil	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	material	Plastic	Glass	Ferrous Metal	Non-Ferrous Metal	Paper
3	energy_saved	5774 Kwh	42 Kwh	642 Kwh	14000 Kwh	4100 kWh
4	crude_oil saved	16 barrels	0.12 barrels	1.8 barrels	40 barrels	11 barrels

- Transpose
- Removed first two columns and first row
- Resetting index
- Renaming the columns

As you can see, we have three columns, material, energy_saved, and crude_oil_saved.

In [9]:
```
clean_energy_saved = (
    energy_saved.T.iloc[1:, 2:]
    .reset_index(drop=True)
    .rename(columns={2: "material", 3: "energy_saved", 4: "crude_oil_saved"})
)
clean_energy_saved
```

Out[9]:

	material	energy_saved	crude_oil_saved
0	Plastic	5774 Kwh	16 barrels
1	Glass	42 Kwh	0.12 barrels
2	Ferrous Metal	642 Kwh	1.8 barrels
3	Non-Ferrous Metal	14000 Kwh	40 barrels
4	Paper	4100 kWh	11 barrels

In [10]:
```
clean_waste_03_17 = waste_03_17.loc[
    :,
    [
        "waste_type",
        "total_waste_generated_tonne",
        "total_waste_recycled_tonne",
        "recycling_rate",
        "year",
    ],
]
```

Let's add recycling rate into our DataFrame as we will be using it later to analysis.

Data Analysis

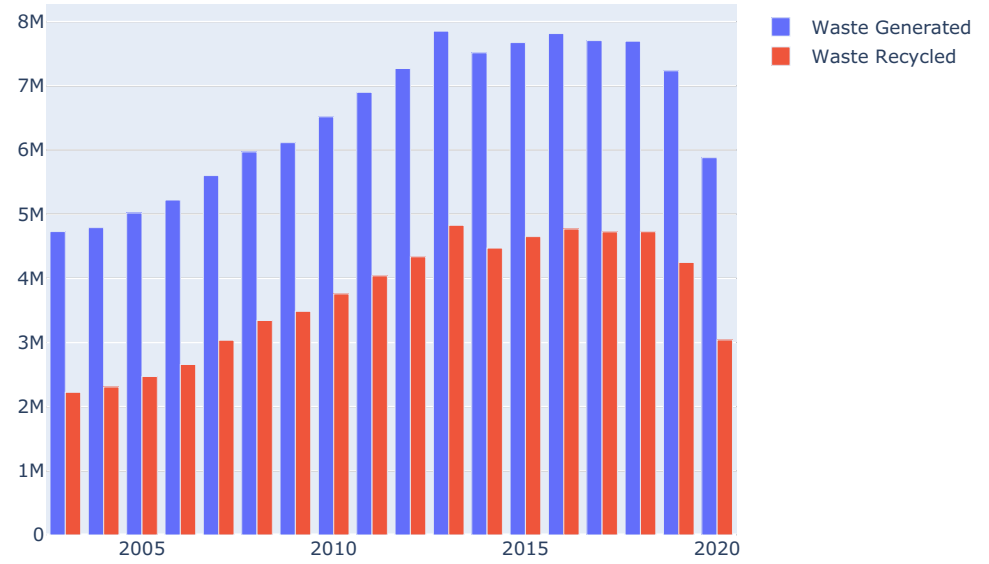
In [12]:
```
data = pd.concat([clean_waste_18_20, clean_waste_03_17]).sort_values(by="year")
overall = data[(data["waste_type"] == "Overall") | (data["waste_type"] == "Total")]


fig = go.Figure()

fig.add_trace(
    go.Bar(
        x=overall["year"],
        y=overall["total_waste_generated_tonne"],
        name="Waste Generated",
    )
)

fig.add_trace(
    go.Bar(
        x=overall["year"],
        y=overall["total_waste_recycled_tonne"],
        name="Waste Recycled",
    )
)

fig.show()
```



In [13]:
```
overall
```

Out[13]:

	waste_type	total_waste_generated_tonne	total_waste_recycled_tonne	year	recycling_rate
209	Total	4728200	2223200.0	2003	0.47
194	Total	4789700	2307100.0	2004	0.48
179	Total	5018200	2469400.0	2005	0.49
164	Total	5220500	2656900.0	2006	0.51
149	Total	5600800	3034800.0	2007	0.54
134	Total	5970200	3342600.0	2008	0.56
119	Total	6114100	3485200.0	2009	0.57
104	Total	6517000	3757500.0	2010	0.58
89	Total	6898300	4038800.0	2011	0.59
74	Total	7269500	4335600.0	2012	0.60
59	Total	7851500	4825900.0	2013	0.61
44	Total	7514500	4471100.0	2014	0.60
29	Total	7673500	4649700.0	2015	0.61
14	Total	7814200	4769000.0	2016	0.61
224	Total	7704300	4724300.0	2017	0.61
14	Overall	7695000	4726000.0	2018	0.61
29	Overall	7234000	4247000.0	2019	0.59
44	Overall	5880000	3040000.0	2020	0.52

In [14]:
```
data['waste_type'].value_counts()
```

Out[14]:
```
Glass                                    18
Paper/Cardboard                          18
Textile/Leather                          18
Plastics                                 17
Total                                    15
Scrap Tyres                              13
Used Slag                                13
Others (stones, ceramics & rubber etc)   12
Sludge                                   11
Non-ferrous Metals                       11
Horticultural Waste                      11
Construction Debris                      11
Food waste                               11
Wood/Timber                              11
Ferrous Metal                             9
Food                                      7
Wood                                      7
Ferrous metal                             5
Used slag                                 5
Ash & Sludge                              5
Scrap tyres                               5
Horticultural waste                       4
Ferrous Metals                            4
Non-ferrous metal                         4
C&D                                       3
Horticultural                             3
Overall                                   3
Others (stones, ceramic, rubber, ect)     2
Construction& Demolition                  2
Non-Ferrous Metal                         2
Others                                    1
Construction & Demolition                 1
Others (stones, ceramics, etc.)           1
Non-ferrous metals                        1
Others (stones, ceramic, rubber, etc.)    1
Plastic                                   1
Ash and sludge                            1
Construction debris                       1
Others (stones, ceramics & rubber etc.)   1
Ash & sludge                              1
Name: waste_type, dtype: int64
```

In [15]:
```
data["waste_type"] = data["waste_type"].str.replace(
    "Non-ferrous metal", "Non-Ferrous Metal"
    )
data["waste_type"] = data["waste_type"].str.replace(
    "Non-ferrous metals", "Non-Ferrous Metal"
    )
data["waste_type"] = data["waste_type"].str.replace(
    "Non-Ferrous Metals", "Non-Ferrous Metal"
    )
data["waste_type"] = data["waste_type"].str.replace(
    "Plastics", "Plastic"
    )
```

```
data["waste_type"] = data["waste_type"].str.replace(
    "Ferrous metal", "Ferrous Metal"
    )
data["waste_type"] = data["waste_type"].str.replace(
    "Paper/Cardboard", "Paper"
    )
```

In [16]:
```
data['waste_type'].value_counts()
```

Out[16]:
```
Plastic                                 18
Glass                                   18
Textile/Leather                         18
Paper                                   18
Total                                   15
Ferrous Metal                           14
Scrap Tyres                             13
Used Slag                               13
Others (stones, ceramics & rubber etc)  12
Wood/Timber                             11
Food waste                              11
Construction Debris                     11
Horticultural Waste                     11
Non-ferrous Metals                      11
Sludge                                  11
Food                                     7
Wood                                     7
Non-Ferrous Metal                        7
Used slag                                5
Ash & Sludge                             5
Scrap tyres                              5
Ferrous Metals                           4
Horticultural waste                      4
Overall                                  3
Horticultural                            3
C&D                                      3
Others (stones, ceramic, rubber, ect)    2
Construction& Demolition                 2
Construction debris                      1
Ash and sludge                           1
Others (stones, ceramic, rubber, etc.)   1
Others (stones, ceramics & rubber etc.)  1
Others                                   1
Others (stones, ceramics, etc.)          1
Construction & Demolition                1
Ash & sludge                             1
Name: waste_type, dtype: int64
```

In [17]:
```
clean_energy_saved
```

Out[17]:

	material	energy_saved	crude_oil_saved
0	Plastic	5774 Kwh	16 barrels
1	Glass	42 Kwh	0.12 barrels
2	Ferrous Metal	642 Kwh	1.8 barrels
3	Non-Ferrous Metal	14000 Kwh	40 barrels
4	Paper	4100 kWh	11 barrels

In [18]:
```
total_data = data.merge(
    clean_energy_saved, how="left", left_on="waste_type", right_on="material"
).dropna()

total_data["energy_saved"] = total_data.loc[:, "energy_saved"].str.replace("kWh", "")

total_data["energy_saved"] = (
    total_data.loc[:, "energy_saved"].str.replace("Kwh", "").astype(int)
)

total_data.head()
```

Out[18]:

	waste_type	total_waste_generated_tonne	total_waste_recycled_tonne	year	recycling_rate	material	energy_saved	crude_oil_saved
2	Glass	65500	6200.0	2003	0.09	Glass	42	0.12 barrels
10	Plastic	579900	39100.0	2003	0.07	Plastic	5774	16 barrels
11	Paper	1084700	466200.0	2003	0.43	Paper	4100	11 barrels
25	Plastic	683100	74100.0	2004	0.11	Plastic	5774	16 barrels
26	Paper	1132100	519900.0	2004	0.46	Paper	4100	11 barrels

In [19]:
```
total_data["total_energy_saved"] = (
    total_data.loc[:, "total_waste_recycled_tonne"] * total_data.loc[:, "energy_saved"]
)

total_data.head()
```

Out[19]:

	waste_type	total_waste_generated_tonne	total_waste_recycled_tonne	year	recycling_rate	material	energy_saved	crude_oil_saved	total
2	Glass	65500	6200.0	2003	0.09	Glass	42	0.12 barrels	
10	Plastic	579900	39100.0	2003	0.07	Plastic	5774	16 barrels	
11	Paper	1084700	466200.0	2003	0.43	Paper	4100	11 barrels	
25	Plastic	683100	74100.0	2004	0.11	Plastic	5774	16 barrels	
26	Paper	1132100	519900.0	2004	0.46	Paper	4100	11 barrels	

Visualization

In [23]:
```
total_data.groupby(by=["waste_type"]).mean()[
    "recycling_rate"
    ].to_frame().style.\
    background_gradient(cmap="Pastel1_r", subset=["recycling_rate"])
```

Out[23]:

waste_type	recycling_rate
Ferrous Metal	0.900714
Glass	0.166667
Non-Ferrous Metal	0.942857
Paper	0.498333
Plastic	0.086667

I wanted to check our final data for outliners and patterns. We found out that there was anomaly at year 2018 and to figure it out we have to check our dataset.

In [24]:
```
fig = px.box(total_data, x="year", y="total_waste_recycled_tonne")
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

In [15]:
```
total_data[total_data['year']==2018]
```

Out[15]:

	waste_type	total_waste_generated_tonne	total_waste_recycled_tonne	year	recycling_rate	material	energy_saved	crude_oil_saved	tota
228	Non-Ferrous Metal	171000	170000.0	2018	0.99	Non-Ferrous Metal	14000	40 barrels	
235	Plastic	949000	41000.0	2018	0.04	Plastic	5774	16 barrels	
236	Paper	1054000	586000.0	2018	0.56	Paper	4100	11 barrels	
237	Ferrous Metal	1269000	126000.0	2018	0.10	Ferrous Metal	642	1.8 barrels	
239	Glass	64000	12000.0	2018	0.19	Glass	42	0.12 barrels	

After going through total waste recycled of 2018, we discovered that total waste generated for Ferrous Metal was 126900 tonne but total recycled waste was 126000. As we know the mean recycling rate for Ferrous metal is 90 but it was showing 10 percent which was odd, so I went back to original data on the site and discovered the mistake. We can clearly see in the PDF that entire zero was missing.

In [25]:
```
total_data.loc[237, "total_waste_recycled_tonne"] = 1260000
total_data["total_energy_saved"] = total_data.loc[:, "total_waste_recycled_tonne"] * (
    total_data.loc[:, "energy_saved"]
)

fig = px.box(total_data, x="year", y="total_waste_recycled_tonne")
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

The Box Plot of total energy saved is all over the place as some of the material produce higher energy kWh per metric tonne.

In [26]:
```
fig = px.box(total_data, x="year", y="total_energy_saved")
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

We can interact more with our data and look for patter in multilevel scatter plot. As we can see total energy saved from paper and plastic have significantly reduce in past few years due to government initiative to control the waste produce.

In [27]:
```
fig = px.scatter(
    total_data,
    x="year",
    y="total_energy_saved",
    size="total_waste_recycled_tonne",
    color="material",
    size_max=60,
)
fig.show()
```

In [28]:
```
total_data.energy_saved.value_counts()
```

Out[28]:
```
42       18
5774     18
4100     18
642      14
14000     7
Name: energy_saved, dtype: int64
```

Energy saved per year

its time to calculate energy saved every year from 2003 to 2020 based on five waste types, plastics, paper, glass, ferrous and non-ferrous metal.

- Group by per year
- Summarize and extract total energy saved
- Converting it into Pandas dataframe
- Converting `total_energy_saved` from float to integer

In [29]:
```
annual_energy_savings = pd.DataFrame(
    total_data.groupby(by=["year"]).sum()["total_energy_saved"],
    columns=["total_energy_saved"],
    ).astype({"total_energy_saved": int})
```

In [30]:
```
annual_energy_savings["total_energy_saved"] = (
    round(annual_energy_savings["total_energy_saved"] / 1000000, 2)\
    .astype(str) + " GWh"
)
annual_energy_savings.tail()
```

Out[30]:

year	total_energy_saved
2016	-2147.48 GWh
2017	-2147.48 GWh
2018	-2147.48 GWh
2019	-2147.48 GWh
2020	-2147.48 GWh

Final Thoughts

We have cleaned our data and made sure that it's ready for merging with other datasets. We have also learned how to detect anomalies in datasets and creating new features. This project was simple, but it taught us a lot of things about data cleaning and data visualization.