DATA WRANGLING & VISUALIZING HEALTHCARE DATASETS

WITH PYTHON
BY DR. ALVIN ANG

CONTENTS

I. DATA CLEANSING A HOSPITAL ADMISSIONS DATASET

- The file is here: https://www.alvinang.sg/s/hospital-admissions-by-sector-annual.csv

- https://www.alvinang.sg/s/Data_Cleansing_a_Hospital_Admissions_Dataset_by_Dr_Alvin_Ang.ipynb

A. STEP 1: IMPORT THE DATASET

	A	B	C	D
1	year	level_1	level_2	value
2	1984	Acute Hospitals Admissions	Public	na
3	1984	Acute Hospitals Admissions	Non-public	na
4	1984	Psychiatric Hospitals Admissions	Public	na
5	1984	Psychiatric Hospitals Admissions	Non-public	na
6	1984	Community Hospitals Admissions	Public	na
7	1984	Community Hospitals Admissions	Non-public	na
8	1985	Acute Hospitals Admissions	Public	na
9	1985	Acute Hospitals Admissions	Non-public	na
10	1985	Psychiatric Hospitals Admissions	Public	na
11	1985	Psychiatric Hospitals Admissions	Non-public	na
12	1985	Community Hospitals Admissions	Public	na
13	1985	Community Hospitals Admissions	Non-public	na
14	1986	Acute Hospitals Admissions	Public	na
15	1986	Acute Hospitals Admissions	Non-public	na

Data Cleansing a Hospital Admissions Dataset by Dr Alvin Ang

https://www.alvinang.sg/s/hospital-admissions-by-sector-annual.csv

---

Step 1: Import the Dataset

```
import pandas as pd

hospital = pd.read_csv('https://www.alvinang.sg/s/hospital-admissions-by-sector-annual.csv')

hospital
```

	year	level_1	level_2	value
0	1984	Acute Hospitals Admissions	Public	na
1	1984	Acute Hospitals Admissions	Non-public	na
2	1984	Psychiatric Hospitals Admissions	Public	na
3	1984	Psychiatric Hospitals Admissions	Non-public	na
4	1984	Community Hospitals Admissions	Public	na
...	...	...	...	...
211	2019	Acute Hospitals Admissions	Non-public	134197
212	2019	Psychiatric Hospitals Admissions	Public	9234
213	2019	Psychiatric Hospitals Admissions	Non-public	0
214	2019	Community Hospitals Admissions	Public	10215
215	2019	Community Hospitals Admissions	Non-public	9828

216 rows × 4 columns

B. STEP 2: REPLACE ALL 'NA' WITH NAN

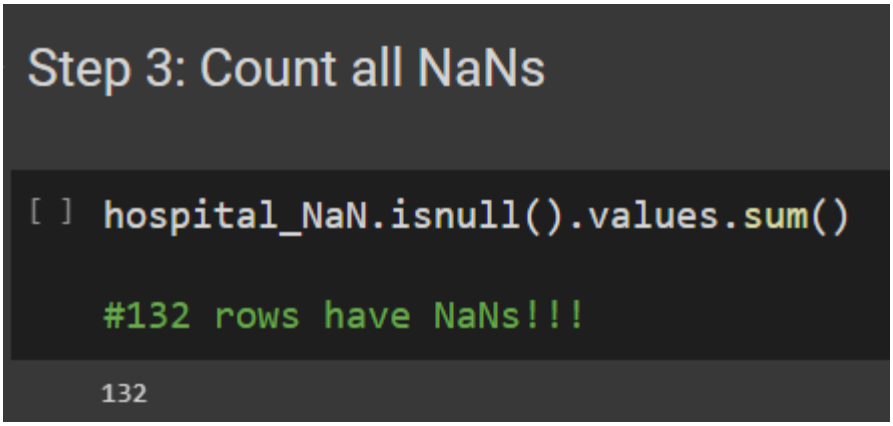
Step 2: Replace all 'na' with NaNs

```
[ ] import numpy as np

    hospital_NaN = health_expenditure.replace('na', np.NaN)

    hospital_NaN
```

	year	level_1	level_2	value
0	1984	Acute Hospitals Admissions	Public	NaN
1	1984	Acute Hospitals Admissions	Non-public	NaN
2	1984	Psychiatric Hospitals Admissions	Public	NaN
3	1984	Psychiatric Hospitals Admissions	Non-public	NaN
4	1984	Community Hospitals Admissions	Public	NaN
...	...	...	...	...
211	2019	Acute Hospitals Admissions	Non-public	134197
212	2019	Psychiatric Hospitals Admissions	Public	9234
213	2019	Psychiatric Hospitals Admissions	Non-public	0
214	2019	Community Hospitals Admissions	Public	10215
215	2019	Community Hospitals Admissions	Non-public	9828

216 rows × 4 columns

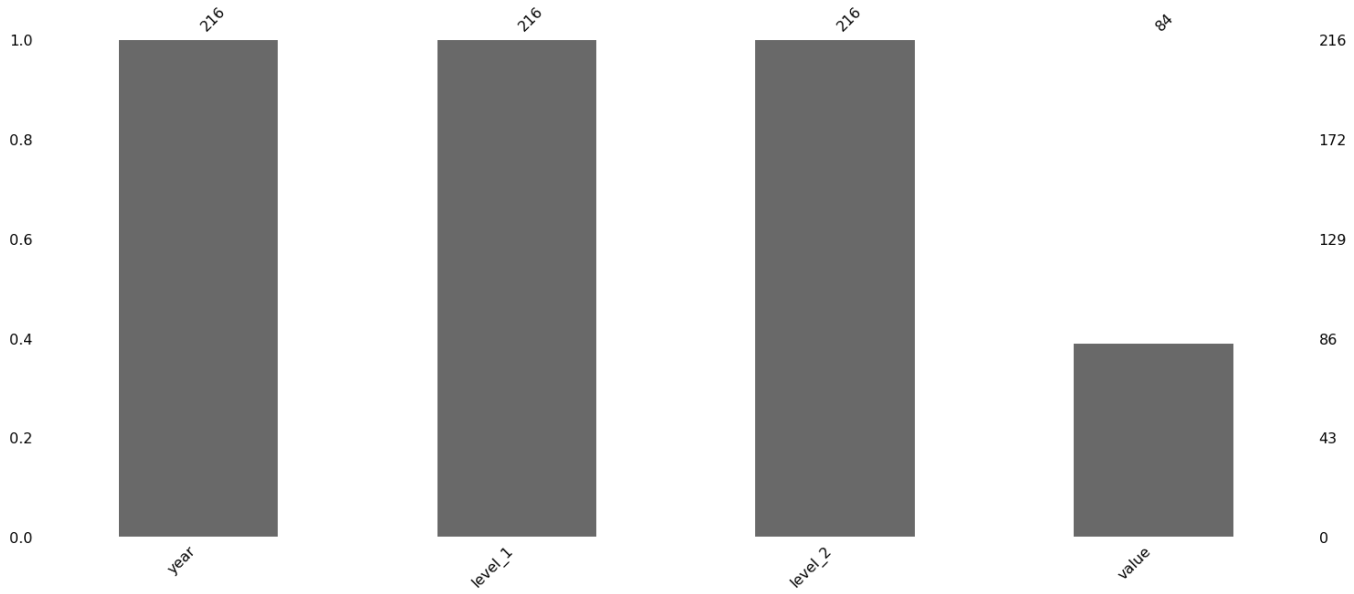
C. STEP 3: COUNT ALL NANS



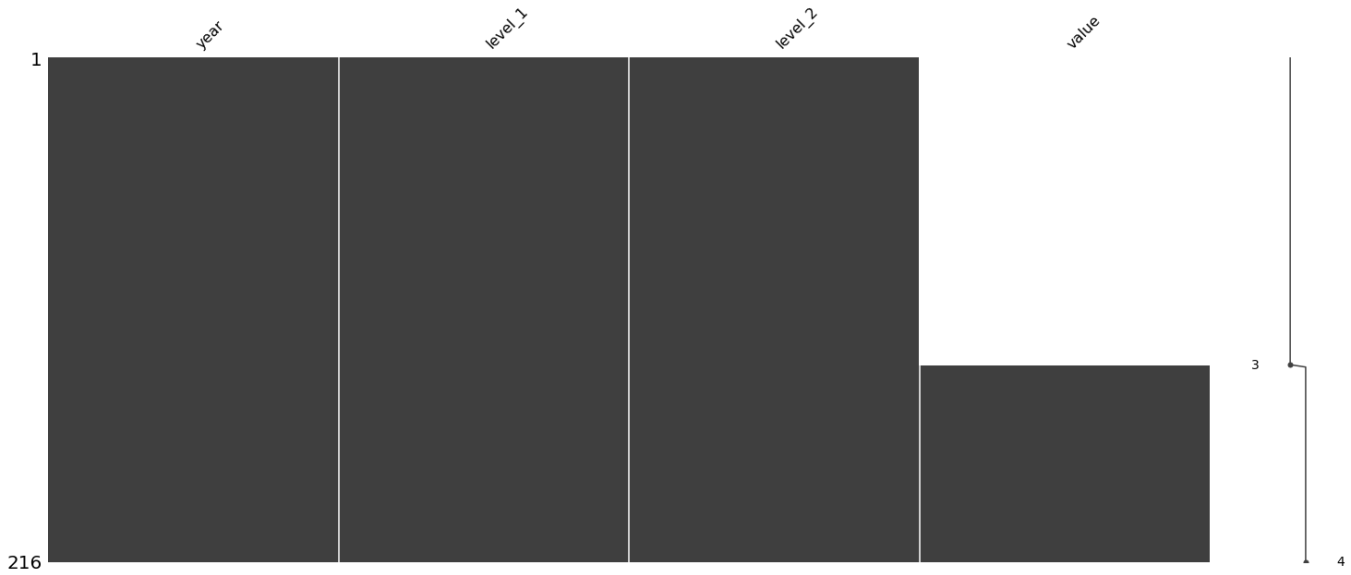
D. STEP 4: USING MISSINGNO TO PREVIEW ALL NANS

Step 4: Using MissingNo to Preview All NaNs

```
import missingno as msno

msno.bar(hospital_NaN)

#seems like the 'value' column has all the NaNs....
# 84 filled rows + 132 NaN rows = 216 rows
```



```
msno.matrix(hospital_NaN)

#'value' column has all the 132 rows of NaNs
```



E. STEP 5: DROP ALL ROWS WITH NANS

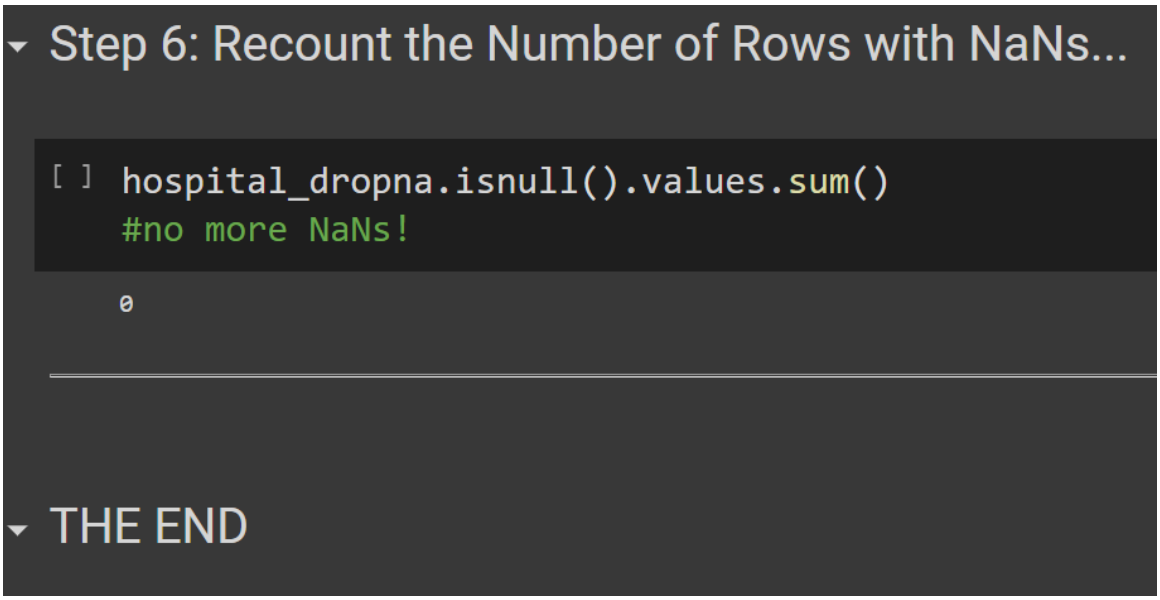
Step 5: Drop All Rows with NaNs

```
[ ] hospital_dropna = hospital_NaN.dropna()
    hospital_dropna

    #all NaNs have been dropped.. we are left with 84 rows
```

	year	level_1	level_2	value
132	2006	Acute Hospitals Admissions	Public	308016
133	2006	Acute Hospitals Admissions	Non-public	92620
134	2006	Psychiatric Hospitals Admissions	Public	8245
135	2006	Psychiatric Hospitals Admissions	Non-public	269
136	2006	Community Hospitals Admissions	Public	0
...	...	...	...	...
211	2019	Acute Hospitals Admissions	Non-public	134197
212	2019	Psychiatric Hospitals Admissions	Public	9234
213	2019	Psychiatric Hospitals Admissions	Non-public	0
214	2019	Community Hospitals Admissions	Public	10215
215	2019	Community Hospitals Admissions	Non-public	9828

84 rows × 4 columns

F. STEP 6: RECOUNT THE NUMBER OF ROWS WITH NANS



II. DATA CLEANSING A HEALTH EXPENDITURE DATASET

- The file can be found here: https://www.alvinang.sg/s/government-health-expenditure.csv

- https://www.alvinang.sg/s/Data_Cleansing_a_Government_Health_Expenditure_Dataset.ipynb

A. STEP 1: IMPORT THE DATA

	A	B	C	D	E
1	financial_year	operating_expenditure	development_expenditure	government_health_expenditure	percentage_gdp
2	2006	1840	96	2009.7	0.9
3	2007	2019	185	2283.2	0.8
4	2008	2379	336	2814.1	1
5	2009	2920	711	3745.8	1.3
6	2010	3258	485	3856.7	1.2
7	2011	3489	453	4091.5	1.2
8	2012	4066	605	4837.3	1.3
9	2013	5044	723	5938.1	1.6
10	2014	5872	1147	7223.1	1.8
11	2015	7520	1413	8639.9	2.1
12	2016	8199	1618	9307	2.1
13	2017	8734	1465	9764.3	2.1
14					

Data Cleansing a Government Health Expenditure Dataset by Dr Alvin Ang

https://www.alvinang.sg/s/government-health-expenditure.csv

Step 1: Import the Data

```
import pandas as pd

health = pd.read_csv('https://www.alvinang.sg/s/government-health-expenditure.csv')

health
```

	financial_year	operating_expenditure	development_expenditure	government_health_expenditure	percentage_gdp
0	2006	1840	96	2009.7	0.9
1	2007	2019	185	2283.2	0.8
2	2008	2379	336	2814.1	1.0
3	2009	2920	711	3745.8	1.3
4	2010	3258	485	3856.7	1.2
5	2011	3489	453	4091.5	1.2
6	2012	4066	605	4837.3	1.3
7	2013	5044	723	5938.1	1.6
8	2014	5872	1147	7223.1	1.8
9	2015	7520	1413	8639.9	2.1
10	2016	8199	1618	9307.0	2.1
11	2017	8734	1465	9764.3	2.1

B. STEP 2: SLICE OUT FINANCIAL YEAR / OPERATING / DEVELOPMENT / GOVERNMENT EXPENDITURE COLUMNS

Step 2: Slice out Financial Year / Operating / Development / Government Expenditure Columns

```
health_sample = pd.read_csv('https://www.alvinang.sg/s/government-health-expenditure.csv',
                            index_col='financial_year',
                            usecols=['financial_year',
                                     'operating_expenditure',
                                     'development_expenditure',
                                     'government_health_expenditure'])
health_sample
```

financial_year	operating_expenditure	development_expenditure	government_health_expenditure
2006	1840	96	2009.7
2007	2019	185	2283.2
2008	2379	336	2814.1
2009	2920	711	3745.8
2010	3258	485	3856.7
2011	3489	453	4091.5
2012	4066	605	4837.3
2013	5044	723	5938.1
2014	5872	1147	7223.1
2015	7520	1413	8639.9
2016	8199	1618	9307.0
2017	8734	1465	9764.3

C. STEP 3: EXPORT TO CSV

Step 3: Export to CSV

```
[ ] health_sample.to_csv('Heath_Expenditure.csv')
```

D. CHECK THE INFO

```
health_sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12 entries, 2006 to 2017
Data columns (total 3 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   operating_expenditure           12 non-null     int64  
 1   development_expenditure         12 non-null     int64  
 2   government_health_expenditure   12 non-null     float64
dtypes: float64(1), int64(2)
memory usage: 384.0 bytes
```

E. STEP 4: SLICE OUT ONLY OPERATING + DEVELOPMENT EXPENDITURE COLUMNS

Step 4: Slice out only Operating + Developement Expenditure Columns

```
a = health_sample[['operating_expenditure','development_expenditure']]
a
```

financial_year	operating_expenditure	development_expenditure
2006	1840	96
2007	2019	185
2008	2379	336
2009	2920	711
2010	3258	485
2011	3489	453
2012	4066	605
2013	5044	723
2014	5872	1147
2015	7520	1413
2016	8199	1618
2017	8734	1465

F. STEP 5: SLICE OUT YEAR 2016 + 2017 (OPERATING AND DEVELOPMENT EXPENDITURE)

Step 5: Slice Out Year 2016 + 2017 (Operating + Development Expenditure)

```
[9] b = health_sample[['operating_expenditure','development_expenditure']].loc[[2016,2017]]
    b
```

financial_year	operating_expenditure	development_expenditure
2016	8199	1618
2017	8734	1465

G. STEP 6: SLICE OUT ROWS 3 TO 7

Step 6: Slice out Rows 3 to 7

```
[10] c = health_sample[['operating_expenditure','development_expenditure']].iloc[3:8]
     c
```

financial_year	operating_expenditure	development_expenditure
2009	2920	711
2010	3258	485
2011	3489	453
2012	4066	605
2013	5044	723

H. STEP 7: SLICE OUT OPERATING EXPENDITURE >5000

Step 7: Slice out Operating Expenditure > 5000

```
[12] d = health_sample[health_sample.operating_expenditure>5000]
     d
```

financial_year	operating_expenditure	development_expenditure	government_health_expenditure
2013	5044	723	5938.1
2014	5872	1147	7223.1
2015	7520	1413	8639.9
2016	8199	1618	9307.0
2017	8734	1465	9764.3

s

I. STEP 8: SLICE OUT BETWEEN 5000 < OPERATING EXPENDITURE < 8000

Step 8: Slice Out Between 5000 < Operating Expenditure < 8000

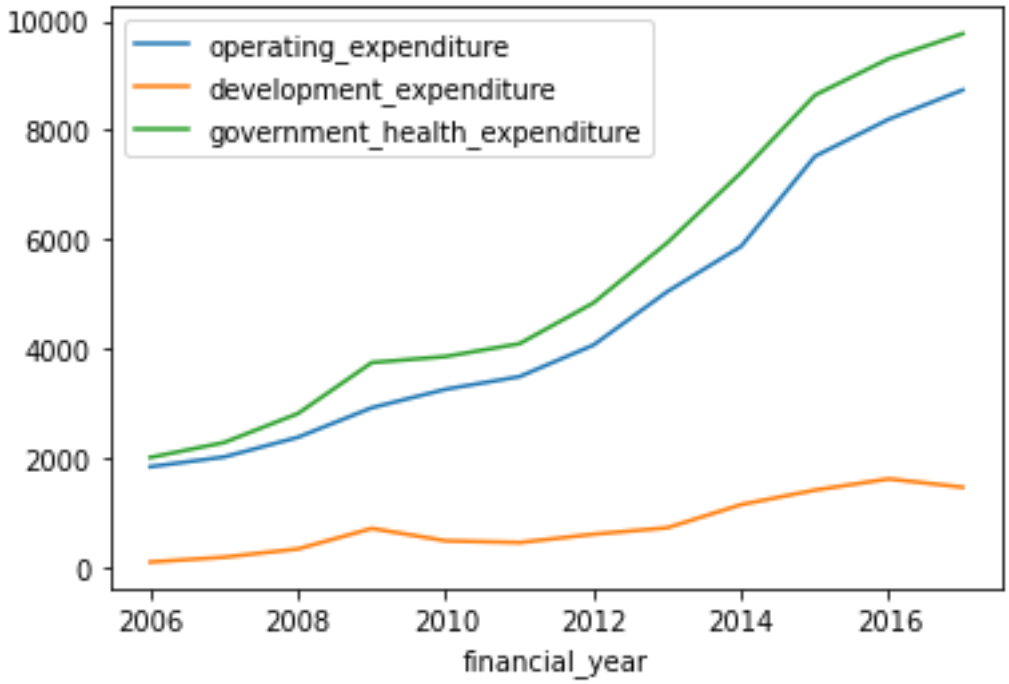
```
[15] e = health_sample[(health_sample.operating_expenditure>5000)
                       & (health_sample.operating_expenditure<8000) ]

     e
```

financial_year	operating_expenditure	development_expenditure	government_health_expenditure
2013	5044	723	5938.1
2014	5872	1147	7223.1
2015	7520	1413	8639.9

J. STEP 9: PLOT EXPENDITURE OVER THE YEARS

Step 9: Plot Expenditure Over the Years

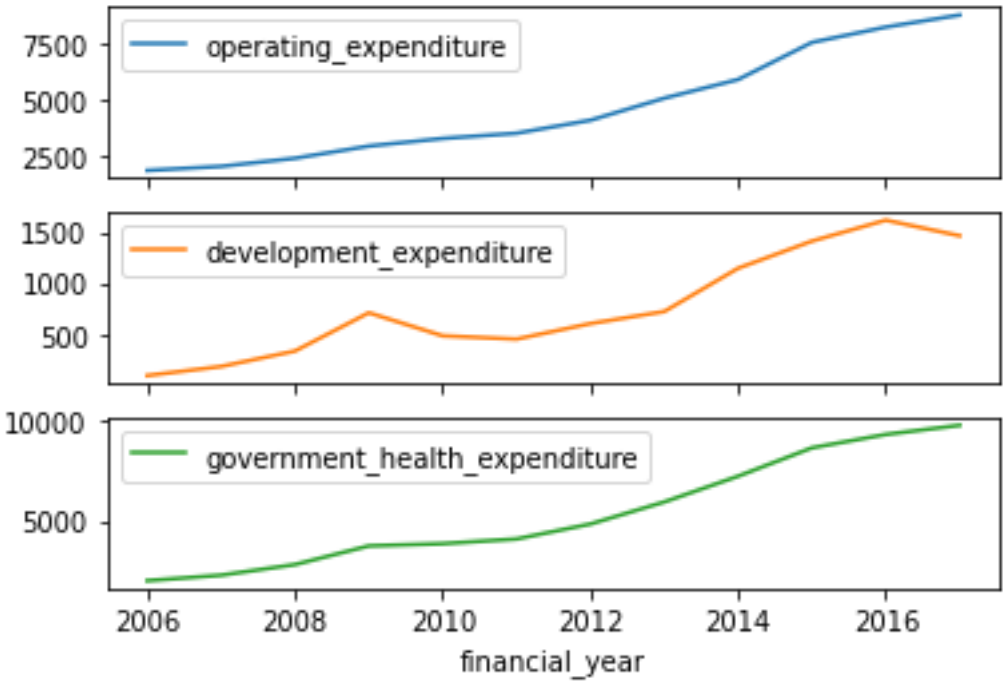
```
health_sample.plot(figsize = (20, 15))
```



K. STEP 10: 3 SUB PLOTS FOR EXPENDITURES

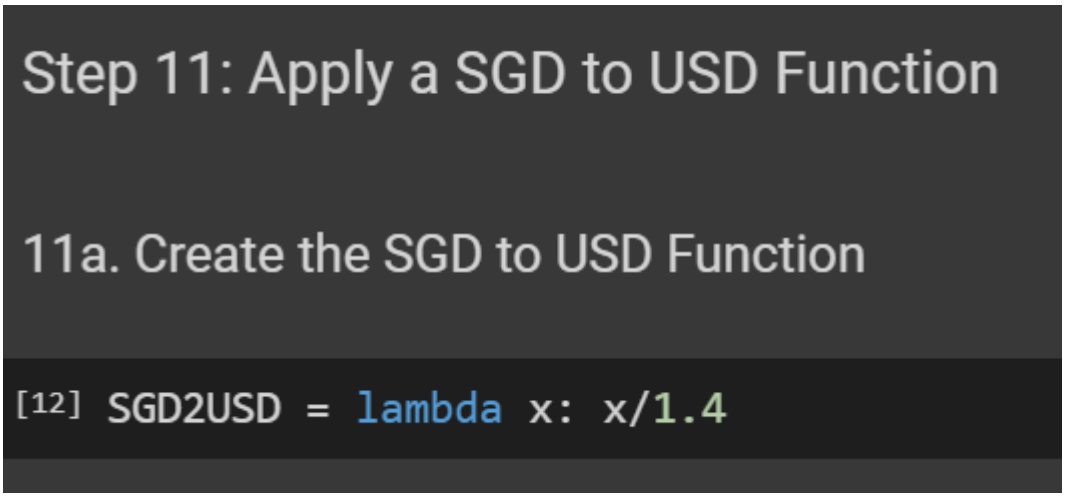
Step 10: 3 Sub Plots for Expenditures

```
health_sample.plot(subplots=True, figsize = (20, 15))
```



L. STEP 11: APPLY A SGD TO USD FUNCTION

1. CREATE THE SGD TO USD FUNCTION



2. APPLY THE SGD TO USD FUNCTION ONTO THE DATASET

11b. Apply the SGD to USD Function onto the Dataset

```
health[['operating_expenditure','development_expenditure','government_health_expenditure']].apply(SGD2USD)
#we select only 3 columns to apply: Operating / Development / Government
```

	operating_expenditure	development_expenditure	government_health_expenditure
0	1314.285714	68.571429	1435.500000
1	1442.142857	132.142857	1630.857143
2	1699.285714	240.000000	2010.071429
3	2085.714286	507.857143	2675.571429
4	2327.142857	346.428571	2754.785714
5	2492.142857	323.571429	2922.500000
6	2904.285714	432.142857	3455.214286
7	3602.857143	516.428571	4241.500000
8	4194.285714	819.285714	5159.357143
9	5371.428571	1009.285714	6171.357143
10	5856.428571	1155.714286	6647.857143
11	6238.571429	1046.428571	6974.500000

M. STEP 12: LINEAR REGRESSION – REGRESSING OPERATING EXPENDITURE TO GOVERNMENT EXPENDITURE

Step 12: Linear Regression - Regressing Operating Expenditure to Government Expenditure

```
[14] x = health_sample[['operating_expenditure','government_health_expenditure']]
     x
```

financial_year	operating_expenditure	government_health_expenditure
2006	1840	2009.7
2007	2019	2283.2
2008	2379	2814.1
2009	2920	3745.8
2010	3258	3856.7
2011	3489	4091.5
2012	4066	4837.3
2013	5044	5938.1
2014	5872	7223.1
2015	7520	8639.9
2016	8199	9307.0
2017	8734	9764.3

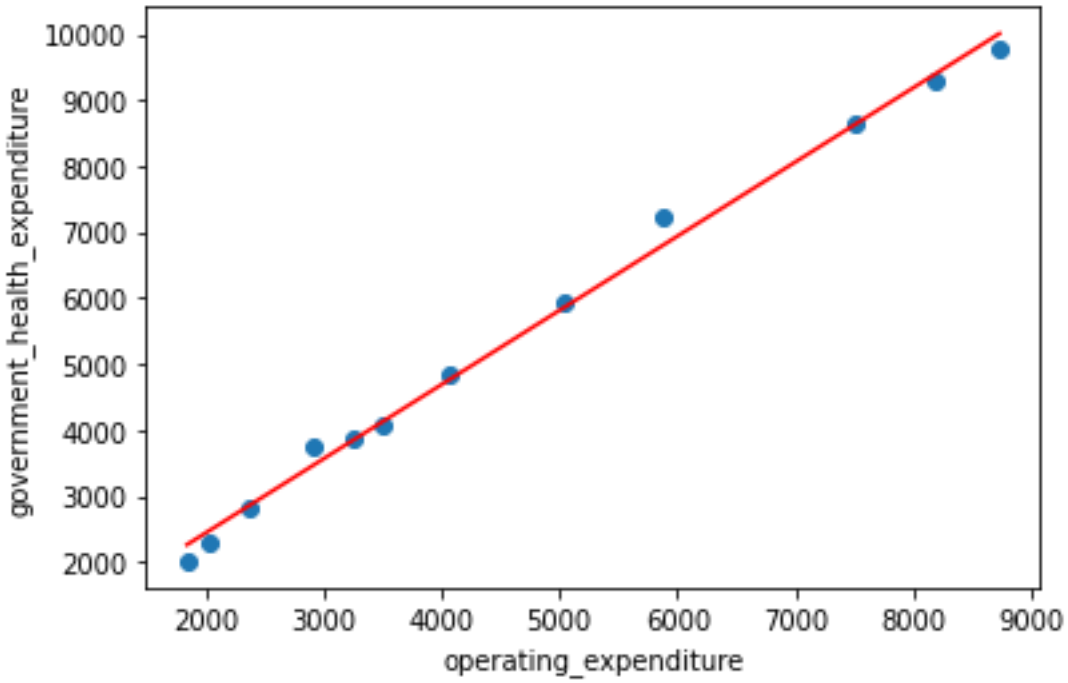
```
[15] y = x.pop('government_health_expenditure')
     y

     # x ~ Operating Expenditure
     # y ~ Government Health Expenditure
```

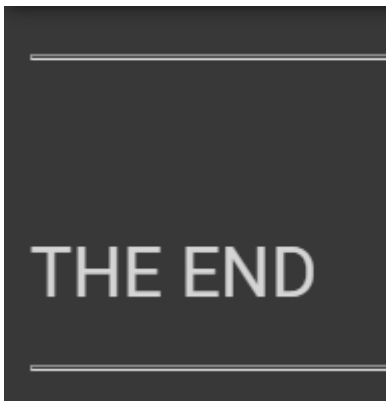
```
financial_year
2006    2009.7
2007    2283.2
2008    2814.1
2009    3745.8
2010    3856.7
2011    4091.5
2012    4837.3
2013    5938.1
2014    7223.1
2015    8639.9
2016    9307.0
2017    9764.3
Name: government_health_expenditure, dtype: float64
```

```
[16] from sklearn import linear_model
     lm = linear_model.LinearRegression()
```

```
[17] lm.fit(x,y)
     yhat = lm.predict(x)
```

```
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

figure(figsize=(20, 15), dpi=80)
plt.scatter(x,y)
plt.xlabel('operating_expenditure')
plt.ylabel('government_health_expenditure')
plt.plot(x,yhat,'r')
plt.show()

#as Operating Expenditure increases, Government Expenditure increases linearly
#together with it
```





III. DATA WRANGLING A LONG-TERM CARE FACILITIES DATASET

https://www.alvinang.sg/s/number-of-residential-long-term-care-facilities-sector-breakdown.csv

https://www.alvinang.sg/s/Data_Wrangling_a_Long_Term_Care_Facilities_Dataset_by_Dr_Alvin_Ang.ipynb

A. STEP 1: IMPORT DATASET

Data Wrangling a Long Term Care Facilities Dataset by Dr Alvin Ang

https://www.alvinang.sg/s/number-of-residential-long-term-care-facilities-sector-breakdown.csv

Step 1: Import Dataset

```
[1] import pandas as pd

    LTC = pd.read_csv('https://www.alvinang.sg/s/number-of-residential-long-term-care-facilities-sector-breakdown.csv')
    LTC.head()
```

	year	type	sector	count
0	2006	Nursing Homes	Public	0
1	2006	Nursing Homes	Not-For-Profit	31
2	2006	Nursing Homes	Private	31
3	2006	Inpatient Hospices	Public	0
4	2006	Inpatient Hospices	Not-For-Profit	2

B. STEP 2: GROUPBY SECTOR

1. USING GROUPBY

Step 2: Groupby Sector

2a. Using Groupby

```
[3] LTC_sector = LTC.groupby(['year','sector']).sum()

    LTC_sector
```

		count
year	sector	
2006	Not-For-Profit	33
	Private	31
	Public	0
2007	Not-For-Profit	33
	Private	31
	Public	0
2008	Not-For-Profit	32
	Private	33
	Public	0
2009	Not-For-Profit	32

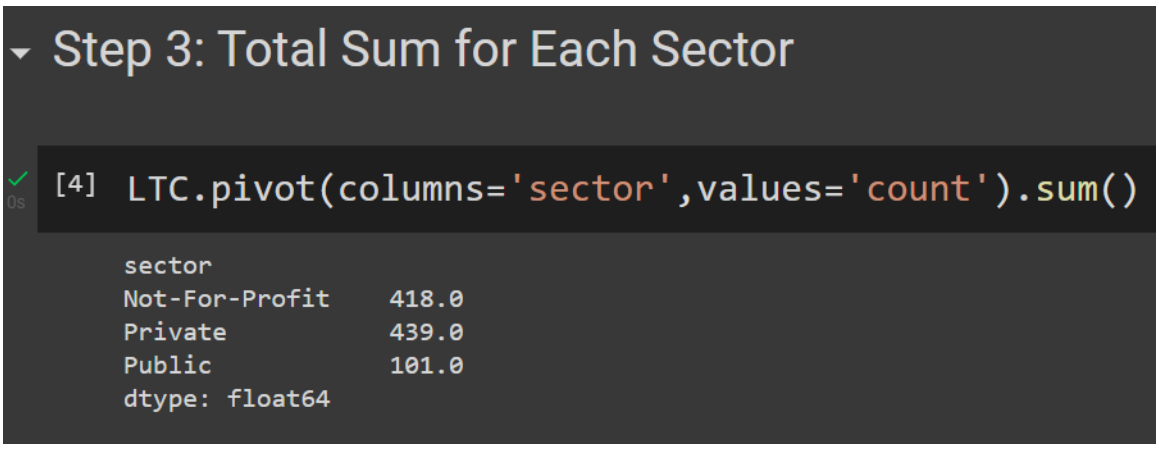
2. USING PIVOT TABLE

2b. Using Pivot Table

```
[5] LTC.pivot_table(index=['year','sector'],values='count',aggfunc='sum')
```

		count
year	sector	
2006	Not-For-Profit	33
	Private	31
	Public	0
2007	Not-For-Profit	33
	Private	31
	Public	0
2008	Not-For-Profit	32
	Private	33
	Public	0
2009	Not-For-Profit	32
	Private	30
	Public	0
2010	Not-For-Profit	32
	Private	32
	Public	0
2011	Not-For-Profit	33
	Private	32

C. STEP 3: TOTAL SUM FOR EACH SECTOR

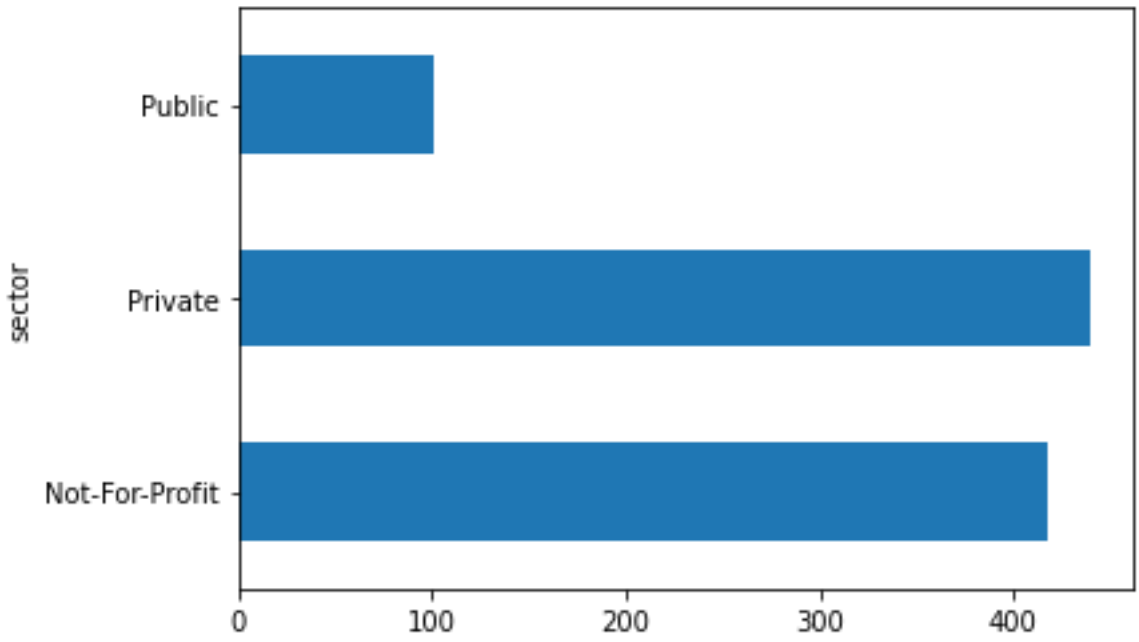


D. STEP 4: HORIZONTAL BAR PLOT FOR THE LONG TERM CARE DATA

Step 4: Horizontal Bar Plot for the Long Term Care Data

- Create a horizontal bar plot of the total long term care facilities by the sector

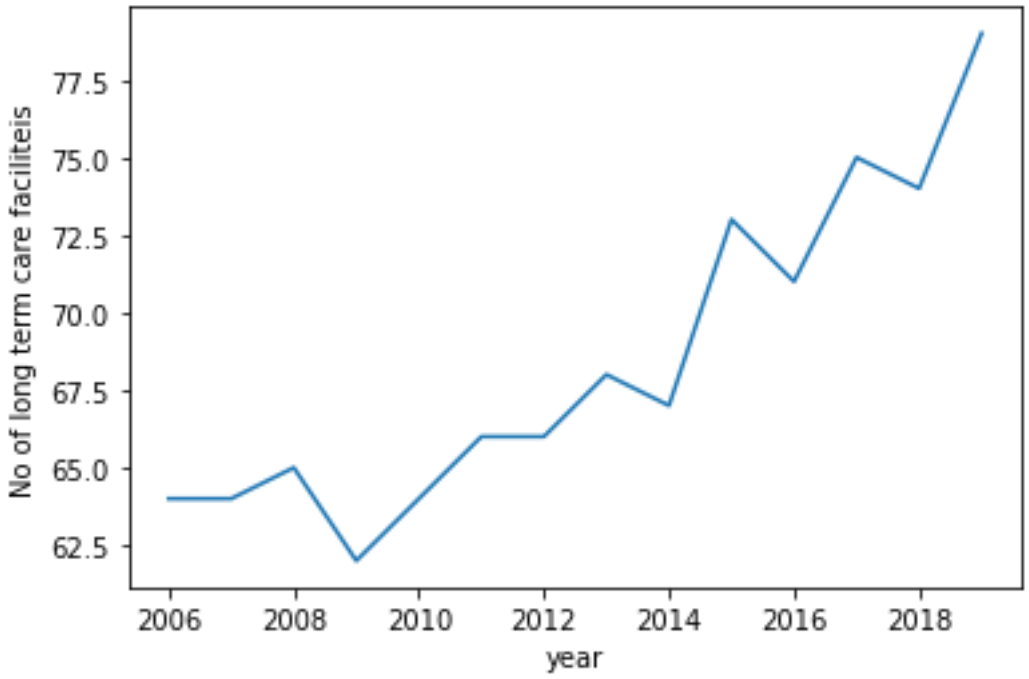
```
[12] from matplotlib.pyplot import figure

     figure(figsize=(20, 15), dpi=80)

     LTC.pivot(columns='sector',values='count').sum().plot.barh()

     # sector
     # Not-For-Profit    418.0
     # Private           439.0
     # Public            101.0
```



E. STEP 5: LINE PLOT NUMBER OF FACILITIES VS YEAR

Step 5: Line Plot Number of Facilities vs Year

```
[ ] import matplotlib.pyplot as plt

    LTC_pivot = LTC.pivot(columns='year',values='count').sum()
    LTC_pivot.plot(figsize = (20, 15))
    plt.xlabel('year')
    plt.ylabel('No of long term care faciliteis')
```



F. STEP 6: USING PIPE TO FILTER OUT NURSING HOMES > 2010

1. LOAD DATA

Step 6: Using Pipe to Filter Out Nursing Homes > 2010

6a. Load Data

```
[7] def load_data():
        return pd.read_csv('https://www.alvinang.sg/s/number-of-residential-long-term-care-facilities-sector-breakdown.csv')

    df = load_data()
```

```
df.head()
```

	year	type	sector	count
0	2006	Nursing Homes	Public	0
1	2006	Nursing Homes	Not-For-Profit	31
2	2006	Nursing Homes	Private	31
3	2006	Inpatient Hospices	Public	0
4	2006	Inpatient Hospices	Not-For-Profit	2

2. FILTER OUT ONLY 'NURSING HOMES'

6b. Filter Out Only 'Nursing Homes'

- df[df['type'] == 'Nursing Homes']

```
[9] #Create a Nursing Homes Filter Function

    def nursinghomes(df):
        df_nursinghomes = df[df['type'] == 'Nursing Homes']

        return df_nursinghomes
```

```
[10] nursinghome_pipe =(
         load_data()
         .pipe(nursinghomes)
     )

     nursinghome_pipe

     #pipe with only 1 funtion --> Nursing Homes Filter
```

	year	type	sector	count
0	2006	Nursing Homes	Public	0
1	2006	Nursing Homes	Not-For-Profit	31
2	2006	Nursing Homes	Private	31
6	2007	Nursing Homes	Public	0
7	2007	Nursing Homes	Not-For-Profit	31
8	2007	Nursing Homes	Private	31
12	2008	Nursing Homes	Public	0
13	2008	Nursing Homes	Not-For-Profit	30
14	2008	Nursing Homes	Private	33
18	2009	Nursing Homes	Public	0

3. FILTER OUT > 2010

6c. Filter Out > 2010

- df[df['year'] > 2010]

```
[11] #Create a Function to Filter Out > 2010

     def above2010(df):
         df_above2010 = df[df['year'] > 2010]

         return df_above2010
```

```
[12] nursinghome_pipe =(
         load_data()
         .pipe(nursinghomes)
         .pipe(above2010)
     )

     nursinghome_pipe

     #pipe with 2 functions --> Nursing Homes above 2010
```

	year	type	sector	count
30	2011	Nursing Homes	Public	1
31	2011	Nursing Homes	Not-For-Profit	31
32	2011	Nursing Homes	Private	32
36	2012	Nursing Homes	Public	1
37	2012	Nursing Homes	Not-For-Profit	31
38	2012	Nursing Homes	Private	32
42	2013	Nursing Homes	Public	5
43	2013	Nursing Homes	Not-For-Profit	28
44	2013	Nursing Homes	Private	33
48	2014	Nursing Homes	Public	6

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.