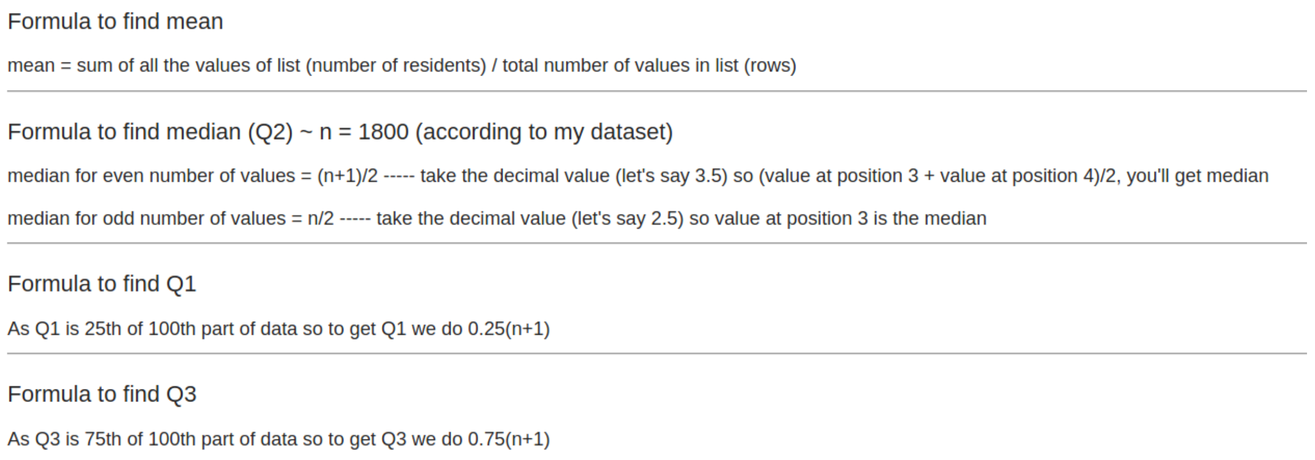
Probability and Statistics – Fall 2020

Momina Atif Dar
P18-0030
Section: B

Boxplot Assignment

Link for dataset:
https://data.gov.sg/dataset/resident-population-by-ethnicity-gender-and-age-group

Screenshots:



Formula to find mean

mean = sum of all the values of list (number of residents) / total number of values in list (rows)

Formula to find median (Q2) ~ n = 1800 (according to my dataset)

median for even number of values = (n+1)/2 ----- take the decimal value (let's say 3.5) so (value at position 3 + value at position 4)/2, you'll get median

median for odd number of values = n/2 ----- take the decimal value (let's say 2.5) so value at position 3 is the median

Formula to find Q1

As Q1 is 25th of 100th part of data so to get Q1 we do 0.25(n+1)

Formula to find Q3

As Q3 is 75th of 100th part of data so to get Q3 we do 0.75(n+1)

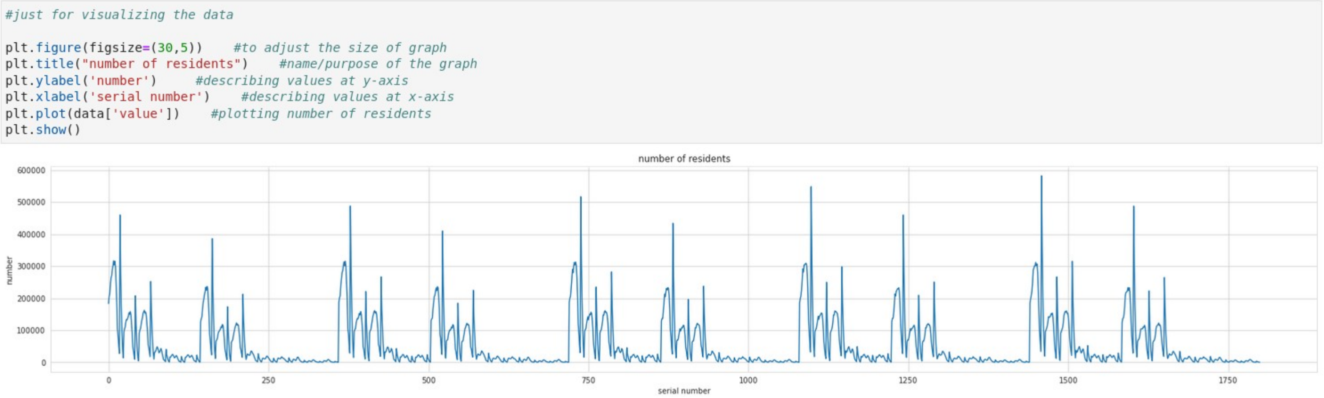
```
#inlcuding all the needed libraries

import pandas as pd
import numpy as np
from scipy import stats
import statistics
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#data reading from file

data = pd.read_csv('./singapore-residents-by-ethnic-group-and-sex-end-june-annual.csv')
```

```
data
```

	year	level_1	level_2	value
0	2015	Total Residents	0 - 4 Years	183575
1	2015	Total Residents	5 - 9 Years	204452
2	2015	Total Residents	10 - 14 Years	214388
3	2015	Total Residents	15 - 19 Years	242902
4	2015	Total Residents	20 - 24 Years	264127
...	...	...	...	...
1795	2019	Other Ethnic Groups (Females)	70 Years & Over	2197
1796	2019	Other Ethnic Groups (Females)	75 Years & Over	1348
1797	2019	Other Ethnic Groups (Females)	80 Years & Over	858
1798	2019	Other Ethnic Groups (Females)	85 Years & Over	454
1799	2019	Other Ethnic Groups (Females)	90 Years & Over	190

1800 rows × 4 columns



```
#just for visualizing the data

plt.figure(figsize=(30,5))    #to adjust the size of graph
plt.title("number of residents")    #name/purpose of the graph
plt.ylabel('number')     #describing values at y-axis
plt.xlabel('serial number')    #describing values at x-axis
plt.plot(data['value'])    #plotting number of residents
plt.show()
```

Calculating mean

```
#calculating mean of 'number of residents' with pandas

data['value'].mean()
```

```
57126.43555555555
```

```
#calculating mean of 'number of residents' with numpy

np.mean(data['value'])
```

```
57126.43555555555
```

Calculating Median

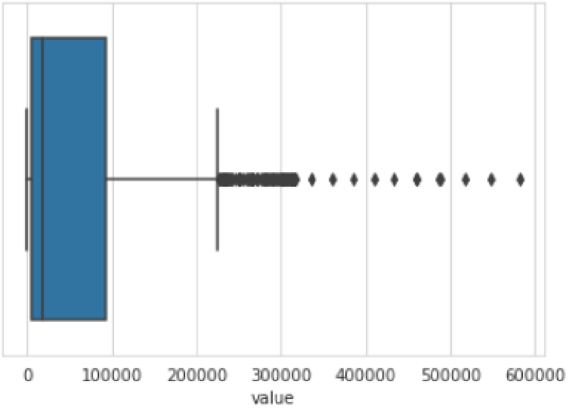
```
#calculating median of 'number of residents' with pandas

data['value'].median()
```

```
17776.0
```

```
#calculating median of 'number of residents' with numpy

np.median(data['value'])
```

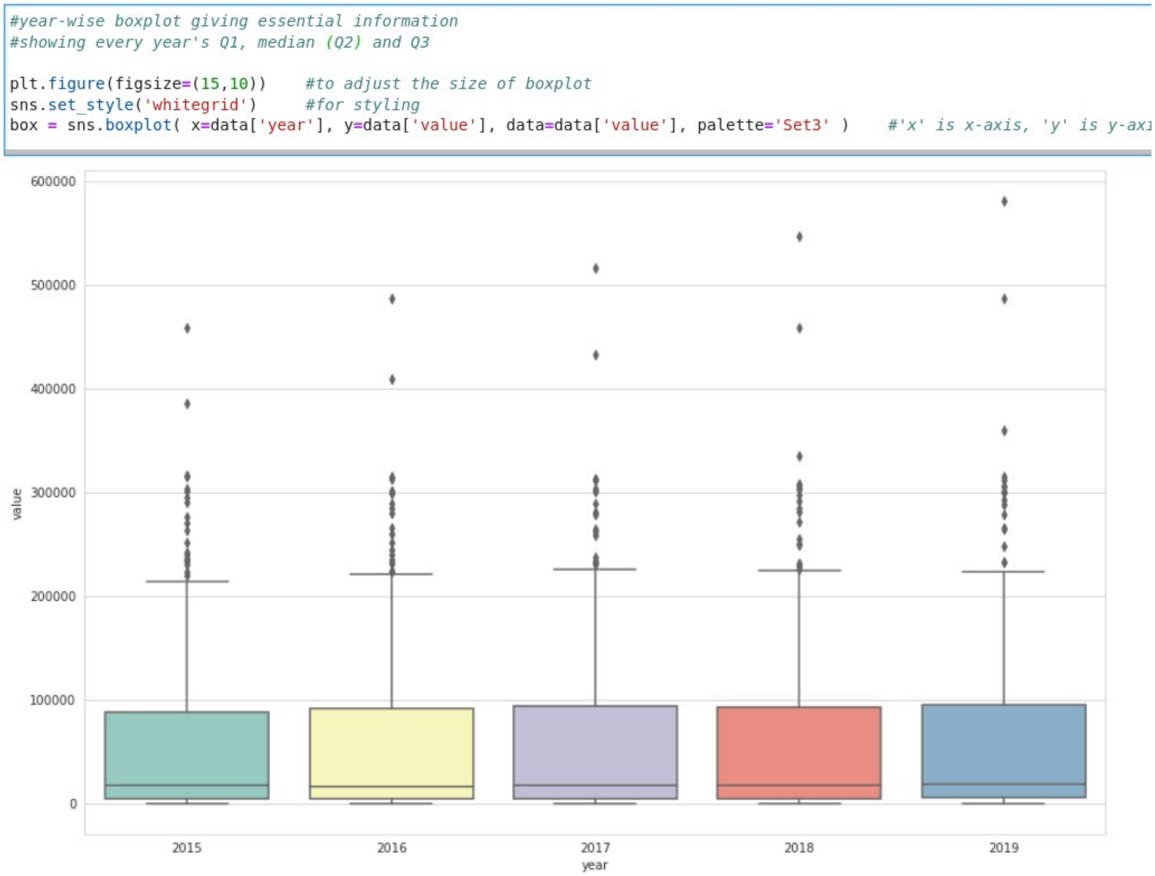
```
17776.0
```

```
#boxplot of all the number of residents as a whole - showing Q1, Q2 and Q3
#it's giving very limited info - can't see the trend happening in years

box = sns.boxplot(x=data['value'])    #'x' is x-axis, number of residents will be displayed on x-axis
```

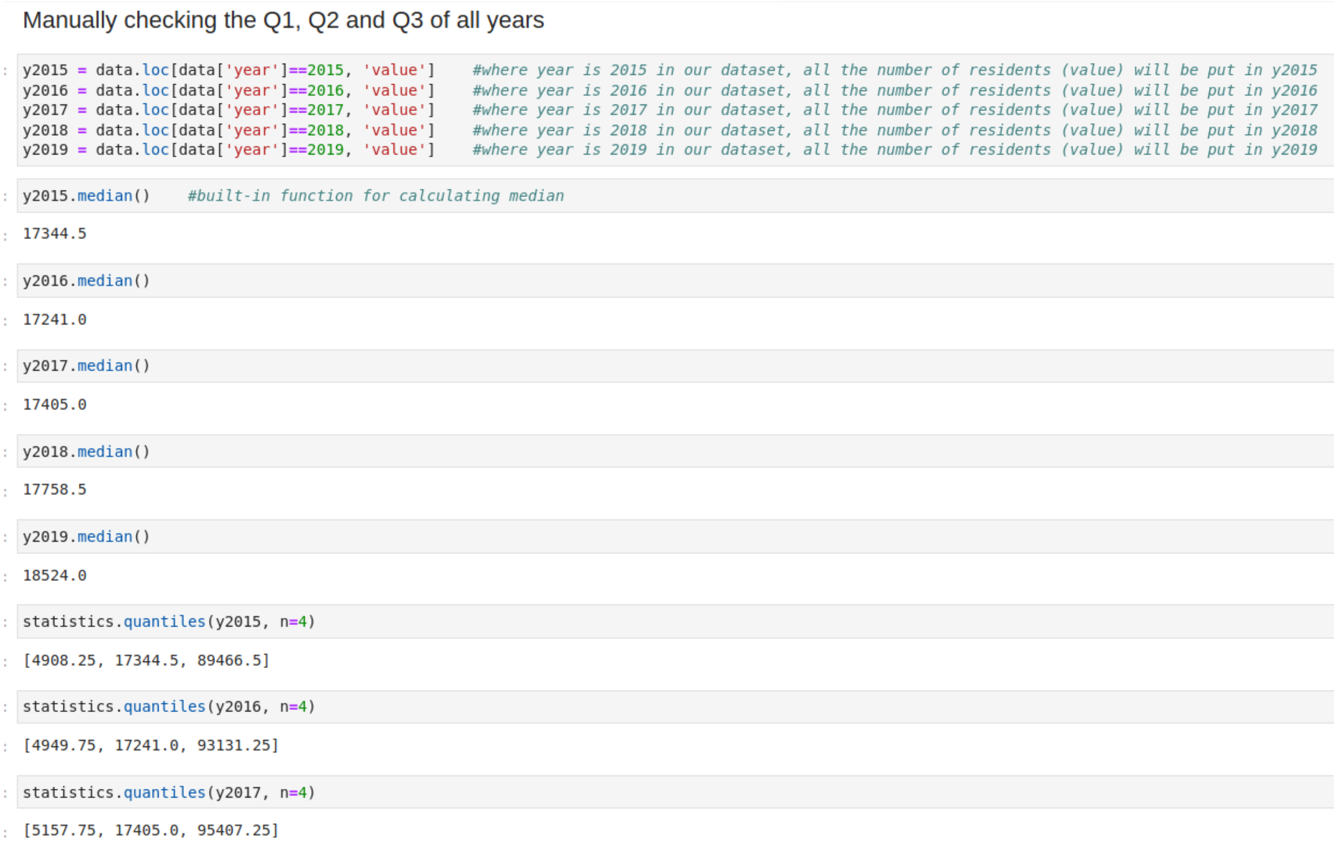


```
#manually checking the Q1, Q2, Q3 of above boxplot

statistics.quantiles(data['value'], n=4)
```

```
[5276.75, 17776.0, 93491.5]
```



```
#year-wise boxplot giving essential information
#showing every year's Q1, median (Q2) and Q3

plt.figure(figsize=(15,10))    #to adjust the size of boxplot
sns.set_style('whitegrid')     #for styling
box = sns.boxplot( x=data['year'], y=data['value'], data=data['value'], palette='Set3' )    #'x' is x-axis, 'y' is y-axi
```



Manually checking the Q1, Q2 and Q3 of all years

```
y2015 = data.loc[data['year']==2015, 'value']    #where year is 2015 in our dataset, all the number of residents (value) will be put in y2015
y2016 = data.loc[data['year']==2016, 'value']    #where year is 2016 in our dataset, all the number of residents (value) will be put in y2016
y2017 = data.loc[data['year']==2017, 'value']    #where year is 2017 in our dataset, all the number of residents (value) will be put in y2017
y2018 = data.loc[data['year']==2018, 'value']    #where year is 2018 in our dataset, all the number of residents (value) will be put in y2018
y2019 = data.loc[data['year']==2019, 'value']    #where year is 2019 in our dataset, all the number of residents (value) will be put in y2019
```

```
y2015.median()    #built-in function for calculating median
```

```
17344.5
```

```
y2016.median()
```

```
17241.0
```

```
y2017.median()
```

```
17405.0
```

```
y2018.median()
```

```
17758.5
```

```
y2019.median()
```

```
18524.0
```

```
statistics.quantiles(y2015, n=4)
```

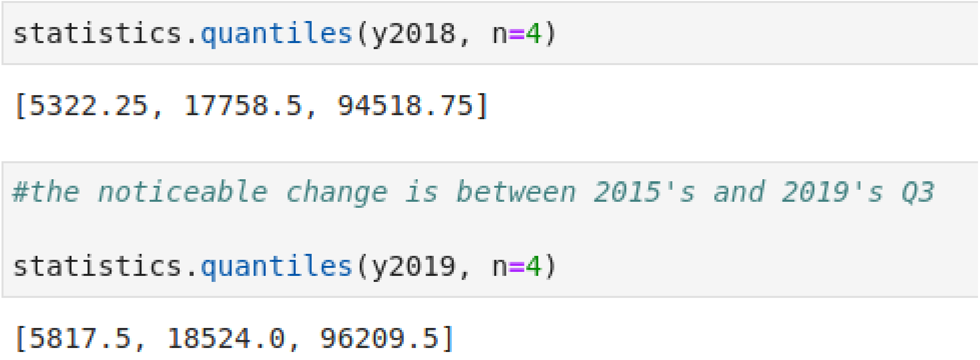
```
[4908.25, 17344.5, 89466.5]
```

```
statistics.quantiles(y2016, n=4)
```

```
[4949.75, 17241.0, 93131.25]
```

```
statistics.quantiles(y2017, n=4)
```

```
[5157.75, 17405.0, 95407.25]
```



```
statistics.quantiles(y2018, n=4)
```

```
[5322.25, 17758.5, 94518.75]
```

```
#the noticeable change is between 2015's and 2019's Q3

statistics.quantiles(y2019, n=4)
```

```
[5817.5, 18524.0, 96209.5]
```