Experiment-1

Title : Apply central tendency and variability on a given dataset

Context: At the end of this activity we will be capable of performing basic analysis on a data set using the measures of: Mean, Mode, Median and the measures of spread: Variance and Standard Deviation. They will also be able to generate the 5 number summary of the data set as well as the Inter Quartile Range (IQR) providing them powerful tools to identify outliers in the data set.

Dataset Description:

The dataset contains 4 CSV files, which covers transactions from 1990 - 2020. It includes features of the flat and sale, such as the year of sale, location of the flat, flat type, street name, block number, area of the flat, lease and resale price.

The approximate floor area includes any recess area purchased, space adding item under HDB's upgrading programmes, roof terrace, etc.

The transactions exclude resale transactions that may not reflect the full market price such as resale between relatives and resale of part shares.

Resale prices should be taken as indicative only as the resale prices agreed between buyers and sellers are dependent on many factors.

Remaining lease is the number of years, months and days left before the lease expires. This information is computed as at the resale flat application and has been rounded up to the nearest month for the purpose of CPF monies usage and HDB loan application.

Prior to March 2012, data is based on date of approval for the resale transactions. For March 2012 onwards, the data is based on date of registration for the resale transactions.

Code & Output:

Using files to mount dataset into Google Colab

```
# using files to mount dataset into Google Colab
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Singapore_Flat_price.csv to Singapore_Flat_price.csv
```

Importing Libraries

```
#importing all libraries
import pandas as pd #for dataframes
import io #for mounting the dataset into pandas df
import numpy as np #for numerical calculation on dataframe
```

Extracting data

```
# Extracting data from the dataset
df =
pd.read_csv(io.BytesIO(uploaded["Singapore_Flat_price.csv"]),index_col=0)

df.head(10)
```

```
               town flat_type block        street_name storey_range  \
month
2017-01  ANG MO KIO    2 ROOM   406  ANG MO KIO AVE 10     10 TO 12
2017-01  ANG MO KIO    3 ROOM   108   ANG MO KIO AVE 4     01 TO 03
2017-01  ANG MO KIO    3 ROOM   602   ANG MO KIO AVE 5     01 TO 03
2017-01  ANG MO KIO    3 ROOM   465  ANG MO KIO AVE 10     04 TO 06
2017-01  ANG MO KIO    3 ROOM   601   ANG MO KIO AVE 5     01 TO 03
2017-01  ANG MO KIO    3 ROOM   150   ANG MO KIO AVE 5     01 TO 03
2017-01  ANG MO KIO    3 ROOM   447  ANG MO KIO AVE 10     04 TO 06
2017-01  ANG MO KIO    3 ROOM   218   ANG MO KIO AVE 1     04 TO 06
2017-01  ANG MO KIO    3 ROOM   447  ANG MO KIO AVE 10     04 TO 06
2017-01  ANG MO KIO    3 ROOM   571   ANG MO KIO AVE 3     01 TO 03

         floor_area_sqm      flat_model  lease_commence_date  \
month
2017-01            44.0        Improved                 1979
2017-01            67.0  New Generation                 1978
2017-01            67.0  New Generation                 1980
2017-01            68.0  New Generation                 1980
2017-01            67.0  New Generation                 1980
2017-01            68.0  New Generation                 1981
2017-01            68.0  New Generation                 1979
2017-01            67.0  New Generation                 1976
2017-01            68.0  New Generation                 1979
2017-01            67.0  New Generation                 1979

            remaining_lease  resale_price
month
2017-01  61 years 04 months      232000.0
2017-01  60 years 07 months      250000.0
2017-01  62 years 05 months      262000.0
2017-01   62 years 01 month      265000.0
2017-01  62 years 05 months      265000.0
2017-01            63 years      275000.0
2017-01  61 years 06 months      280000.0
2017-01  58 years 04 months      285000.0
2017-01  61 years 06 months      285000.0
2017-01  61 years 04 months      285000.0
```

Finding Mean,Median and Mode

```
# Finding all mean values in the dataframe
print("Mean Values in the Distrubution")
print(df.mean())
print()
```

```
# Finding all median values of integer data in df
print("Meadian values in the Distrubution")
print(df.median())
print()

# Finding the mode of all integer data in df
print("Mode")
print(df.mode())
print()

# FInding the standard deviation of all integer data in df
print("Standard deviation")
print(df.std())
print()
```

```
Mean Values in the Distrubution
floor_area_sqm             97.768362
lease_commence_date      1994.444175
resale_price           444886.900540
dtype: float64

Meadian values in the Distrubution
floor_area_sqm             95.0
lease_commence_date      1995.0
resale_price           415000.0
dtype: float64

Mode
       town flat_type block     street_name storey_range  floor_area_sqm
\
0  SENGKANG    4 ROOM     2  YISHUN RING RD     04 TO 06            67.0

  flat_model  lease_commence_date     remaining_lease  resale_price
0    Model A                 1985  94 years 09 months      400000.0

Standard deviation
floor_area_sqm             24.263575
lease_commence_date        13.064066
resale_price           154824.263389
dtype: float64

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  This is separate from the ipykernel package so we can avoid doing
imports until
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:18:
```

```
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
```

Finding Interquartile Range

```
# Finding interquartile range
# finding quater 3 and quater 1 [75 percentile and 25 percerntile]

#finding the iqr of the field "floor_area_sqm"
q3,q1 = np.percentile(df["floor_area_sqm"],[75,25])
iqr = q3-q1
print("Inter quartile range:",iqr)

#finding the iqr of the field "resale_price"
q3,q1 = np.percentile(df["resale_price"],[75,25])
iqr = q3-q1
print("Inter quartile range:",iqr)
```

```
Inter quartile range: 31.0
Inter quartile range: 187000.0
```

Finding iqr of every column

```
#defining a function to find iqr of every column in the dataframe
print("Inter quartile range")
def calIQR(x):
  return np.subtract(*np.percentile(x,[75,25]))

print(df[["floor_area_sqm","resale_price"]].apply(calIQR))
```

```
Inter quartile range
floor_area_sqm        31.0
resale_price      187000.0
dtype: float64
```