Assignment 1

Name:Pratiksha Gaikwad
Gr.No: 21920026
RollNo: 321066
TY-Comp   Batch: A3

Aim : Perform the following operations using **Python** on suitable data sets, Read a data from different formats(like csv,xls),indexing and selecting data, sort data, describe attributes of data, checking data types of each column, counting unique values of data, format of each column, converting variable data type (e.g. from long to short, vise versa), identifying missing values and fill in the missing values.

Outcome:

Dataset Name: World Happiness Report 2019
Overview of Data :
Each column of data has the next description.

1. **Country (region)** : Name of the country.
2. **Ladder:** is a measure of life satisfaction.
3. **SD of Ladder:** Standard deviation of the ladder.
4. **Positive affect:** Measure of positive emotion.
5. **Negative affect:** Measure of negative emotion.
6. **Social support** :The extent to which Social support contributed to the calculation of the Happiness Score.
7. **Freedom:** The extent to which Freedom contributed to the calculation of the Happiness Score.
8. **Corruption:** The extent to which Perception of Corruption contributes to Happiness Score.
9. **Generosity:** The extent to which Generosity contributed to the calculation of the Happiness Score.
10. **Log of GDP per capita:** The extent to which GDP contributes to the calculation of the Happiness Score.
11. **Healthy life expectancy:** The extent to which Life expectancy contributed to the calculation of the Happiness Score.

WhatsApp 21920026 - Google Drive Documents/data science 21920026_Assignment_1

localhost:8888/notebooks/Documents/data%20science/ass%201/21920026_Assignment_1%20.ipynb

jupyter 21920026_Assignment_1 Last Checkpoint: 09/22/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

```python
In [1]: import pandas as pd
        import numpy as np
        dta=pd.read_csv("/home/pratiksha/Documents/data science/ass 1/world-happiness-report-2019.csv")
        dta
```

Out[1]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
0	Finland	1	4	41.0	10.0	2.0	5.0	4.0	47.0	22.0	27.0
1	Denmark	2	13	24.0	26.0	4.0	6.0	3.0	22.0	14.0	23.0
2	Norway	3	8	16.0	29.0	3.0	3.0	8.0	11.0	7.0	12.0
3	Iceland	4	9	3.0	3.0	1.0	7.0	45.0	3.0	15.0	13.0
4	Netherlands	5	1	12.0	25.0	15.0	19.0	12.0	7.0	12.0	18.0
...	...	...	...	...	...	...	...	...	...	...	...
151	Rwanda	152	63	54.0	102.0	144.0	21.0	2.0	90.0	132.0	103.0
152	Tanzania	153	122	78.0	50.0	131.0	78.0	34.0	49.0	125.0	118.0
153	Afghanistan	154	25	152.0	133.0	151.0	155.0	136.0	137.0	134.0	139.0
154	Central African Republic	155	117	132.0	153.0	155.0	133.0	122.0	113.0	152.0	150.0
155	South Sudan	156	140	127.0	152.0	148.0	154.0	61.0	85.0	140.0	143.0

156 rows × 11 columns

```python
In [2]: dta.head()
```

Out[2]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy

---

WhatsApp 21920026 - Google Drive Documents/data science 21920026_Assignment_1

localhost:8888/notebooks/Documents/data%20science/ass%201/21920026_Assignment_1%20.ipynb

jupyter 21920026_Assignment_1 Last Checkpoint: 09/22/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

```python
In [2]: dta.head()
```

Out[2]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
0	Finland	1	4	41.0	10.0	2.0	5.0	4.0	47.0	22.0	27.0
1	Denmark	2	13	24.0	26.0	4.0	6.0	3.0	22.0	14.0	23.0
2	Norway	3	8	16.0	29.0	3.0	3.0	8.0	11.0	7.0	12.0
3	Iceland	4	9	3.0	3.0	1.0	7.0	45.0	3.0	15.0	13.0
4	Netherlands	5	1	12.0	25.0	15.0	19.0	12.0	7.0	12.0	18.0

```python
In [3]: dta.tail()
```

Out[3]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
151	Rwanda	152	63	54.0	102.0	144.0	21.0	2.0	90.0	132.0	103.0
152	Tanzania	153	122	78.0	50.0	131.0	78.0	34.0	49.0	125.0	118.0
153	Afghanistan	154	25	152.0	133.0	151.0	155.0	136.0	137.0	134.0	139.0
154	Central African Republic	155	117	132.0	153.0	155.0	133.0	122.0	113.0	152.0	150.0
155	South Sudan	156	140	127.0	152.0	148.0	154.0	61.0	85.0	140.0	143.0

```python
In [4]: #data selecting and indexing
        ser = pd.Series(dta['Country (region)'])
        data = ser.head(10)
        data
```

Out[4]: 0   Finland

```
In [4]: #data selecting and indexing
        ser = pd.Series(dta['Country (region)'])
        data = ser.head(10)
        data

Out[4]: 0        Finland
        1        Denmark
        2         Norway
        3        Iceland
        4    Netherlands
        5    Switzerland
        6         Sweden
        7    New Zealand
        8         Canada
        9        Austria
        Name: Country (region), dtype: object

In [5]: data[3:7]

Out[5]: 3        Iceland
        4    Netherlands
        5    Switzerland
        6         Sweden
        Name: Country (region), dtype: object

In [6]: data[6:]

Out[6]: 6         Sweden
        7    New Zealand
        8         Canada
        9        Austria
        Name: Country (region), dtype: object
```

```
In [9]: #sorting data
        dta.sort_values(["Country (region)"], axis=0,ascending=True,inplace=True)
        dta

Out[9]:
```

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
153	Afghanistan	154	25	152.0	133.0	151.0	155.0	136.0	137.0	134.0	139.0
106	Albania	107	126	90.0	108.0	133.0	87.0	134.0	60.0	81.0	40.0
87	Algeria	88	56	113.0	106.0	101.0	149.0	46.0	128.0	72.0	78.0
46	Argentina	47	97	28.0	93.0	46.0	54.0	109.0	123.0	55.0	37.0
115	Armenia	116	82	126.0	145.0	117.0	123.0	93.0	129.0	91.0	64.0
...	...	...	...	...	...	...	...	...	...	...	...
107	Venezuela	108	141	77.0	135.0	49.0	145.0	110.0	139.0	78.0	71.0
93	Vietnam	94	27	121.0	27.0	64.0	23.0	86.0	97.0	105.0	49.0
150	Yemen	151	85	153.0	75.0	100.0	147.0	83.0	155.0	141.0	124.0
137	Zambia	138	145	84.0	128.0	115.0	73.0	69.0	53.0	115.0	131.0
145	Zimbabwe	146	123	63.0	34.0	110.0	96.0	63.0	141.0	131.0	129.0

156 rows × 11 columns

```
In [10]: dta.sort_values(["Positive affect"], axis=0,ascending=False,inplace=True) #descending
         dta['Positive affect'].head(20)

Out[10]: 148    155.0
         78     154.0
         150    153.0
         153    152.0
         125    151.0
```

Jupyter 21920026_Assignment_1 Last Checkpoint: 09/22/2020 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 ○

Code

```
In [10]: dta.sort_values(["Positive affect"], axis=0,ascending=False,inplace=True) #descending
         dta['Positive affect'].head(20)
```

```
Out[10]: 148    155.0
         78     154.0
         150    153.0
         153    152.0
         125    151.0
         90     150.0
         80     149.0
         69     148.0
         123    147.0
         136    146.0
         124    145.0
         63     144.0
         72     143.0
         146    142.0
         118    141.0
         83     140.0
         128    139.0
         41     138.0
         99     137.0
         131    136.0
         Name: Positive affect, dtype: float64
```

```
In [11]: dta.sort_values(["Positive affect"], axis=0,ascending=False,inplace=True) #descending
         dta['Positive affect'].tail(10)
```

```
Out[11]: 91      9.0
         26      8.0
         30      7.0
         22      6.0
         104     5.0
         11      4.0
```

Jupyter 21920026_Assignment_1 Last Checkpoint: 09/22/2020 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 ○

Code

```
In [11]: dta.sort_values(["Positive affect"], axis=0,ascending=False,inplace=True) #descending
         dta['Positive affect'].tail(10)
```

```
Out[11]: 91      9.0
         26      8.0
         30      7.0
         22      6.0
         104     5.0
         11      4.0
         3       3.0
         111     2.0
         62      1.0
         28      NaN
         Name: Positive affect, dtype: float64
```

```
In [12]: #describe attributes of data
         dta.describe()
```

Out[12]:

	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
count	156.000000	156.000000	155.000000	155.000000	154.000000	155.000000	148.000000	155.000000	152.000000	150.000000
mean	78.500000	78.500000	78.000000	78.000000	77.902597	78.000000	74.500000	78.000000	76.500000	75.500000
std	45.177428	45.177428	44.888751	44.888751	45.018772	44.888751	42.868014	44.888751	44.022721	43.445368
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	39.750000	39.750000	39.500000	39.500000	39.250000	39.500000	37.750000	39.500000	38.750000	38.250000
50%	78.500000	78.500000	78.500000	78.000000	77.500000	78.000000	74.500000	78.000000	76.500000	75.500000
75%	117.250000	117.250000	116.500000	116.500000	116.750000	116.500000	111.250000	116.500000	114.250000	112.750000
max	156.000000	156.000000	155.000000	155.000000	155.000000	155.000000	148.000000	155.000000	152.000000	150.000000

```
In [13]: #checking data types of each column
         dta.dtypes

Out[13]: Country (region)          object
         Ladder                    int64
         SD of Ladder              int64
         Positive affect           float64
         Negative affect           float64
         Social support            float64
         Freedom                   float64
         Corruption                float64
         Generosity                float64
         Log of GDP\nper capita    float64
         Healthy life\nexpectancy  float64
         dtype: object

In [14]: #counting unique values of data
         unival= dta.nunique()
         unival

Out[14]: Country (region)          156
         Ladder                    156
         SD of Ladder              156
         Positive affect           155
         Negative affect           155
         Social support            154
         Freedom                   155
         Corruption                148
         Generosity                155
         Log of GDP\nper capita    152
         Healthy life\nexpectancy  150
         dtype: int64

In [15]: #format of each column
```

```
In [15]: #format of each column
         dta.describe().transpose()
```

Out[15]:

	count	mean	std	min	25%	50%	75%	max
Ladder	156.0	78.500000	45.177428	1.0	39.75	78.5	117.25	156.0
SD of Ladder	156.0	78.500000	45.177428	1.0	39.75	78.5	117.25	156.0
Positive affect	155.0	78.000000	44.888751	1.0	39.50	78.0	116.50	155.0
Negative affect	155.0	78.000000	44.888751	1.0	39.50	78.0	116.50	155.0
Social support	154.0	77.902597	45.018772	1.0	39.25	77.5	116.75	155.0
Freedom	155.0	78.000000	44.888751	1.0	39.50	78.0	116.50	155.0
Corruption	148.0	74.500000	42.868014	1.0	37.75	74.5	111.25	148.0
Generosity	155.0	78.000000	44.888751	1.0	39.50	78.0	116.50	155.0
Log of GDP\nper capita	152.0	76.500000	44.022721	1.0	38.75	76.5	114.25	152.0
Healthy life\nexpectancy	150.0	75.500000	43.445368	1.0	38.25	75.5	112.75	150.0

```
In [16]: #converting data type
         dta= dta.fillna(0)
         dta['Freedom'] = dta['Freedom'].astype(int)
         dta['Freedom'].dtypes

Out[16]: dtype('int64')

In [17]: dta.dtypes

Out[17]: Country (region)          object
         Ladder                    int64
         SD of Ladder              int64
         Positive affect           float64
```

WhatsApp   21920026 - Google Drive   Documents/data science   21920026_Assignment_1

localhost:8888/notebooks/Documents/data%20science/ass%201/21920026_Assignment_1%20.ipynb

jupyter 21920026_Assignment_1 Last Checkpoint: 09/22/2020 (autosaved)

File   Edit   View   Insert   Cell   Kernel   Widgets   Help   Trusted | Python 3 ○

```
In [17]: dta.dtypes
```

```
Out[17]: Country (region)          object
         Ladder                     int64
         SD of Ladder               int64
         Positive affect          float64
         Negative affect          float64
         Social support           float64
         Freedom                    int64
         Corruption               float64
         Generosity               float64
         Log of GDP\nper capita   float64
         Healthy life\nexpectancy float64
         dtype: object
```

```
In [18]: #identifying missing values and fill in the missing value
         dta=pd.read_csv("/home/pratiksha/Documents/data science/ass 1/world-happiness-report-2019.csv")
         dta.tail(15).isnull()
```

Out[18]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
141	False	False	False	False	False	False	False	False	False	False	False
142	False	False	False	False	False	False	False	False	False	False	False
143	False	False	False	False	False	False	False	False	False	False	False
144	False	False	False	False	False	False	False	False	False	False	False
145	False	False	False	False	False	False	False	False	False	False	False
146	False	False	False	False	False	False	False	False	False	False	False
147	False	False	False	False	False	False	False	False	False	False	False
148	False	False	False	False	False	False	False	False	True	False

---

```
In [19]: dta.isnull().sum()
```

```
Out[19]: Country (region)          0
         Ladder                    0
         SD of Ladder              0
         Positive affect           1
         Negative affect           1
         Social support            2
         Freedom                   1
         Corruption                8
         Generosity                1
         Log of GDP\nper capita    4
         Healthy life\nexpectancy  6
         dtype: int64
```

```
In [20]: dta['Positive affect'].isnull().sum()
```

```
Out[20]: 1
```

```
In [21]: dta.isnull().sum().sum()
```

```
Out[21]: 24
```

```
In [22]: dta.fillna('0', inplace = True)
```

```
In [23]: dta.isnull().sum().sum()
```

```
Out[23]: 0
```

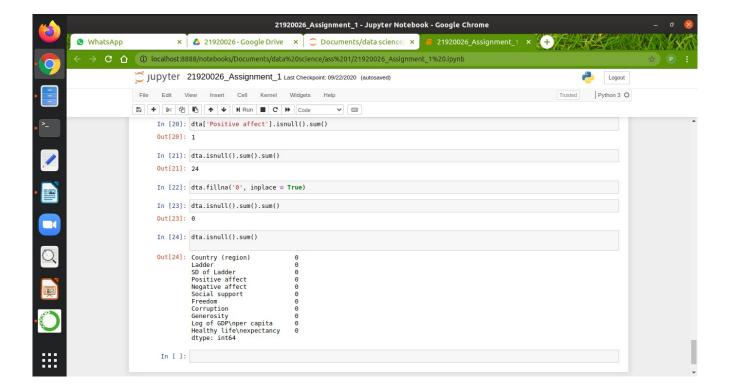```
In [24]: dta.isnull().sum()
```

**Description:**

- As shown in above images by using python read_csv() dataset is accessed.
- By using python indexing and selecting various slicing operations are performed.
- Int64, float64, object is datatypes are there for columns.
- By using python fillna() and isnull() are used for recognizing null entries in dataset.

**Interpretation:**

- There are 156 entries in dataset and finland is happiest country in world.

- There is 24 null values in dataset that are filled with 0 by using fillna().