```
In [77]:  import pandas as pd
          from sklearn.impute import KNNImputer
```

Read the CSV file

```
In [78]:  df = pd.read_csv('../Country Quater Wise Visitors.csv')
          df.tail()
```

Out[78]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	NaN	NaN	NaN	NaN	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

1. Filling with a default value

```
In [79]:  df2 = df.fillna(value = 0)
          df2.tail()
```

Out[79]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	0.0	0.0	0.0	0.0	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

2. Filling with previous value

```
In [80]: df2 = df.fillna(method = 'pad')
         df2.tail()
```

Out[80]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarter (Jan March
58	Japan	28.0	19.7	25.1	27.2	26.8	19.7	22.9	30.6	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

3. Filling with next value

```
In [81]: df3 = df.fillna(method = 'bfill')
         df3.tail()
```

Out[81]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarter (Jan March
58	Japan	28.0	19.7	25.1	27.2	31.1	19.7	23.1	26.1	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

4. Filling with Mean

```
In [82]: def get_dict(dataFrame, ignoreList, strategy):
             dictionary = {}
             for series_name, series in dataFrame.items():
                 if series_name in ignoreList:
                     continue
                 if strategy == "mean":
                     ans = dataFrame[series_name].mean()
                 elif strategy == "median":
                     ans = dataFrame[series_name].median()
                 elif strategy == "mode":
                     ans = dataFrame[series_name].mode().max()
                 elif strategy == "max":
                     ans = dataFrame[series_name].max()
                 elif strategy == "min":
                     ans = dataFrame[series_name].min()
                 dictionary[series_name] = ans
             return dictionary

         df4 = df.fillna(get_dict(df, ["Country of Nationality"], "mean"))
         df4.tail()
```

Out[82]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	20 4 quart (O De
58	Japan	28.0	19.7	25.1	27.2	30.440984	18.598361	21.740984	29.219672	28.6	...	24.62	27.
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.100000	19.700000	23.100000	26.100000	30.1	...	23.87	27.
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.800000	16.900000	22.300000	28.000000	32.6	...	23.75	27.
61	Australia	25.2	17.1	18.4	39.3	25.700000	16.700000	17.600000	40.000000	26.9	...	17.67	39.
62	New Zealand	25.5	16.3	18.3	39.9	25.400000	16.500000	17.900000	40.200000	25.3	...	17.34	39.

5 rows × 29 columns

5. Filling with Median

```
In [83]: df5 = df.fillna(get_dict(df, ["Country of Nationality"], "median"))
         df5.tail()
```

Out[83]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	28.6	18.1	21.9	28.9	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

6. Filling with Mode

```
In [84]: df6 = df.fillna(get_dict(df, ["Country of Nationality"], "mode"))
         df6.tail()
```

Out[84]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	26.3	22.9	19.6	31.9	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

7. Filling with Max

```
In [85]: df7 = df.fillna(get_dict(df, ["Country of Nationality"], "max"))
         df7.tail()
```

Out[85]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	51.5	24.6	43.5	43.7	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

8. Filling with Min

```
In [86]:  df8 = df.fillna(get_dict(df, ["Country of Nationality"], "min"))
          df8.tail()
```

Out[86]:

	Country of Nationality	2014 1st quarter (Jan-March)	2014 2nd quarter (Apr-June)	2014 3rd quarter (July-Sep)	2014 4th quarter (Oct-Dec))	2015 1st quarter (Jan-March)	2015 2nd quarter (Apr-June)	2015 3rd quarter (July-Sep)	2015 4th quarter (Oct-Dec)	2016 1st quarter (Jan-March)	...	2018 3rd quarter (July-Sep)	2018 4th quarter (Oct-Dec)	2019 1s quarte (Jan March
58	Japan	28.0	19.7	25.1	27.2	16.8	10.0	6.4	18.8	28.6	...	24.62	27.44	28.2
59	Korea (Republic Of	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	...	23.87	27.54	31.9
60	Republic Of China Taiwan	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	...	23.75	27.30	30.8
61	Australia	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	...	17.67	39.11	26.6
62	New Zealand	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	...	17.34	39.06	27.6

5 rows × 29 columns

9. Filling with Most Frequent

```
In [87]:  df9 = df.drop("Country of Nationality", axis="columns")
          imp1 = SimpleImputer(strategy="most_frequent")
          res1 = imp1.fit_transform(df9)
          newData1 = pd.DataFrame(res1)
          newData1.tail()
```

Out[87]:

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21	22	23	24	25	26	27
58	28.0	19.7	25.1	27.2	26.3	16.6	19.6	24.6	28.6	20.3	...	24.62	27.44	28.2	20.9	24.2	26.7	93.4	0.5	1.7	4.3
59	32.1	19.1	22.6	26.2	31.1	19.7	23.1	26.1	30.1	19.1	...	23.87	27.54	31.9	20.9	22.0	25.3	81.0	3.2	6.1	9.6
60	29.7	16.8	22.7	30.8	32.8	16.9	22.3	28.0	32.6	16.8	...	23.75	27.30	30.8	16.9	25.0	27.2	97.7	0.0	0.5	1.7
61	25.2	17.1	18.4	39.3	25.7	16.7	17.6	40.0	26.9	15.3	...	17.67	39.11	26.6	17.0	16.9	39.5	95.2	0.4	1.2	3.2
62	25.5	16.3	18.3	39.9	25.4	16.5	17.9	40.2	25.3	14.9	...	17.34	39.06	27.6	17.2	16.6	38.6	94.2	0.8	1.5	3.5

5 rows × 28 columns

10 Filling using KNN

```
In [88]:  df10 = df.drop("Country of Nationality", axis="columns")
          imp2 = KNNImputer(n_neighbors=2)
          res2 = imp2.fit_transform(df10)
          newData2 = pd.DataFrame(res2)
          newData2.tail()
```

Out[88]:

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21	22	23	24	25	26	27
58	28.0	19.7	25.1	27.2	25.6	20.95	24.05	29.4	28.6	20.3	...	24.62	27.44	28.2	20.9	24.2	26.7	93.4	0.5	1.7	4.3
59	32.1	19.1	22.6	26.2	31.1	19.70	23.10	26.1	30.1	19.1	...	23.87	27.54	31.9	20.9	22.0	25.3	81.0	3.2	6.1	9.6
60	29.7	16.8	22.7	30.8	32.8	16.90	22.30	28.0	32.6	16.8	...	23.75	27.30	30.8	16.9	25.0	27.2	97.7	0.0	0.5	1.7
61	25.2	17.1	18.4	39.3	25.7	16.70	17.60	40.0	26.9	15.3	...	17.67	39.11	26.6	17.0	16.9	39.5	95.2	0.4	1.2	3.2
62	25.5	16.3	18.3	39.9	25.4	16.50	17.90	40.2	25.3	14.9	...	17.34	39.06	27.6	17.2	16.6	38.6	94.2	0.8	1.5	3.5

5 rows × 28 columns