Session 11: Missing Data (Solutions Only)

1. What is a NaN Value?

Q1: Predict the output of the each of the following lines without typing them, and then verify by typing.

```
nan==True
nan==False
nan**2
nan!=nan
pd.isnull('null')
pd.isnull(nan==True)
```

Answer: The first two statements are False because nan is not equal to anything.

The third statement evaluates to nan because any numerical operation with nans evaluates to nan.

The fourth statement (perhaps surprisingly) is True, because `nan==nan` is False.

The last two statements are all False. This is because `'null'` is a string and `'nan==True'` evaluates to `False`, both of which are not `nan`.

Q2: Explain the output of the following code using your knowledge of NaNs.

```
[12]: df=pd.DataFrame([[nan,10],[50,nan],[nan,20],[20,10]],columns=['Capacity','Demand'])
      df
```

```
   Capacity  Demand
0       NaN    10.0
1      50.0     NaN
2       NaN    20.0
3      20.0    10.0
```

```
[13]: (df['Capacity']-df['Demand']).sum()
```

```
10.0
```

Explanation: This takes the difference of the two columns first, and only the last item is not null. Hence, the total is 10. (See below)

```
[14]: df['Capacity']-df['Demand']
```

```
0     NaN
1     NaN
2     NaN
3    10.0
dtype: float64
```

```
[15]: df['Capacity'].sum()-df['Demand'].sum()
```

```
30.0
```

Explanation: This takes sums the none-null entries in each column first. The result is 70-40, which is 30.

```
[16]: df.sum(axis=0)
```

```
Capacity    70.0
Demand      40.0
dtype: float64
```

```
[17]: df.sum(axis=1)
```

```
0    10.0
1    50.0
2    20.0
3    30.0
dtype: float64
```

```
[18]: df.sum(axis=1,skipna=False)
```

```
0     NaN
1     NaN
2     NaN
3    30.0
dtype: float64
```

Explanation: The first command sums the columns and the second sums the rows. The default behavior is to skip NA values. The third command does not skip NA values, so the sum for the first three rows is NaN.

2. Handling Missing Values

Loading the Ebola Dataset

Q3-a: Count the number of missing entries in each column of the `df` DataFrame from Q2.

```
[24]: df.isnull().sum()
```

```
Capacity    2
Demand      1
dtype: int64
```

Q3-b: Write a command to count the number of rows in which the difference between capacity and demand is missing.

```
[25]: (df['Capacity']-df['Demand']).isnull().sum()
```

```
3
```

Q3-c: Similar to Q3-b, except count the number of rows in which the difference is not missing.

```
[26]: (df['Capacity']-df['Demand']).count()
```

```
1
```

```
[27]: (df['Capacity']-df['Demand']).notnull().sum()
```

```
1
```

Filling Missing Values

Q4-a: An analyst would like to calculate the average value of the "Cases_Guinea" column of the the ebola Dataset. The analyst runs the below command. Explain why this result might be misleading.

```
[41]: guinea.mean()
```

```
911.0645161290323
```

Q4-b: Write a command that corrects the above issue.

```
[42]: guinea.fillna(method='ffill').mean()
```

```
1023.7295081967213
```

Any of the following also obtains a similar result.

```
[43]: guinea.fillna(method='bfill').mean()
```

```
1005.139344262295
```

```
[44]: guinea.interpolate().mean()
```

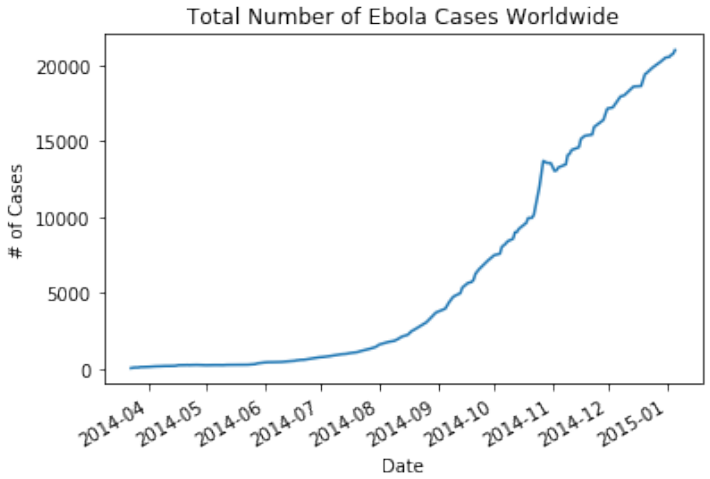
```
1014.4344262295082
```

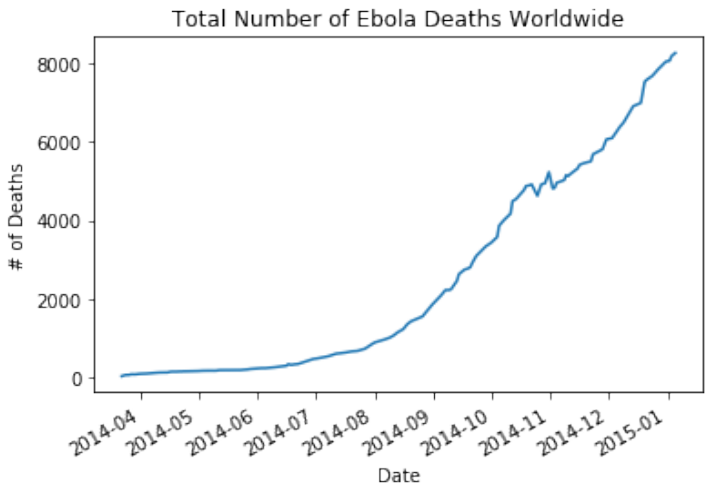
Q5: Plot the total number of cases and the total number of deaths due to Ebola in the data set from all of the countries, while handling missing data appropriately.

```
[45]: cases=ebola.fillna(method='bfill').iloc[:,1:9].sum(axis=1)
      deaths=ebola.fillna(method='bfill').iloc[:,9:].sum(axis=1)

      import matplotlib.pyplot as plt
      cases.plot(title='Total Number of Ebola Cases Worldwide')
      plt.ylabel('# of Cases')
      plt.show()

      deaths.plot(title='Total Number of Ebola Deaths Worldwide')
      plt.ylabel('# of Deaths')
      plt.show()
```





Q6: Obtain the first non-null value of the column "Cases_UnitedStates", as well as the date on which this is recorded. (Since the dates are ordered backward, this is the last recorded value and the last recorded date). Do the same also for the last non-null value of the column.

```
[46]: # Index of first non-null value
      usa=ebola['Cases_UnitedStates']
      usa.first_valid_index()
```

```
Timestamp('2014-12-07 00:00:00')
```

```
[47]: # First non-null value
      usa[usa.first_valid_index()]
```

```
4.0
```

```
[48]: # Index of last non-null value
      usa.last_valid_index()
```

```
Timestamp('2014-10-01 00:00:00')
```

```
[49]: # Last non-null value
      usa[usa.last_valid_index()]
```

```
1.0
```

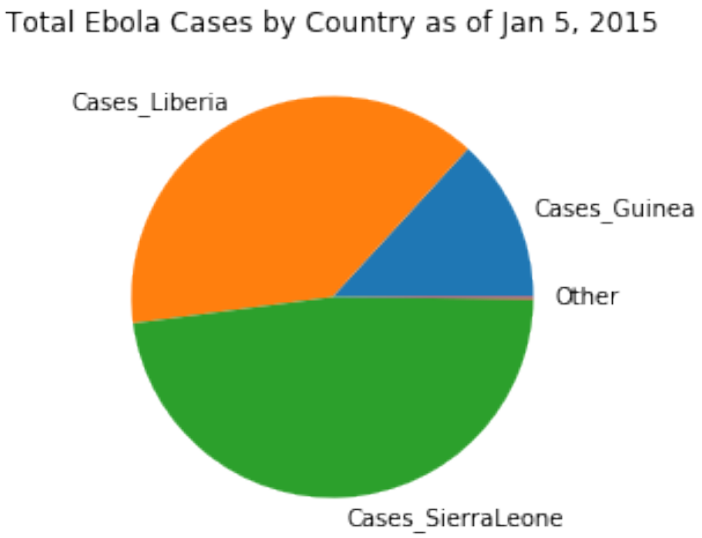
Q7: Based on the information in the dataset, obtain an estimate of the number of cases of ebola in each of the countries in the dataset on Jan 5, 2015. Appropriately display the information in a pie chart.

```
[50]: ebola.fillna(method='bfill').iloc[0,1:9]
```

```
Cases_Guinea           2776.0
Cases_Liberia          8166.0
Cases_SierraLeone     10030.0
Cases_Nigeria            20.0
Cases_Senegal             1.0
Cases_UnitedStates        4.0
Cases_Spain               1.0
Cases_Mali                7.0
Name: 2015-01-05 00:00:00, dtype: float64
```

```
[51]: ebola_cleaned=ebola.fillna(method='bfill')
      result=ebola_cleaned.iloc[0,1:4]
      result['Other']=sum(ebola_cleaned.iloc[0,4:9])

      import matplotlib.pyplot as plt
      result.plot(kind='pie',title='Total Ebola Cases by Country as of Jan 5, 2015')
      plt.ylabel('')
      plt.show()
```



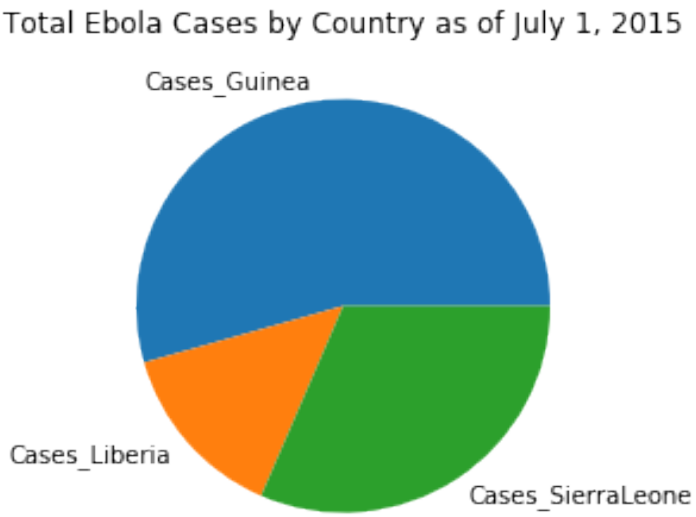
This is how you do it for an arbitrary date that may or may not be in the Dataset. The following example is for 2014 Sep 1.

```
[54]: import numpy as np
      ebola.index[np.where(ebola.index<pd.datetime(2014,7,1))].max()
```

```
Timestamp('2014-06-30 00:00:00')
```

```
[55]: date=ebola.index[np.where(ebola.index<pd.datetime(2014,7,1))].max()
      result=ebola_cleaned.loc[date,:].iloc[1:4]
      result['Other']=sum(ebola_cleaned.loc[date,:].iloc[4:9])

      import matplotlib.pyplot as plt
      result.plot(kind='pie',title='Total Ebola Cases by Country as of July 1, 2015')
      plt.ylabel('')
      plt.show()
```


Total Ebola Cases by Country as of July 1, 2015
Cases_Guinea
Cases_Liberia
Cases_SierraLeone