Data Management - Assignment 8

Data and Package Import

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_excel('data/impurity_dataset-training.xlsx')
```

1. Selecting specific subsets of time data

Create a subset of the Dow dataset containing only data from Dec. 5 - Dec. 12, 2015.
The data should only include columns for `x1` to `x12`. You do not have to rename the columns.
Print the first 5 rows of the new dataset.

In [3]:
```
df_datetime = df.set_index('Date')
df_oneweek = df_datetime['2015-12-05':'2015-12-12']
df_oneweek = df_oneweek[df_oneweek.columns[:12]]
```

In [4]:
```
df_oneweek.head(5)
```

Out[4]:

Date	x1:Primary Column Reflux Flow	x2:Primary Column Tails Flow	x3:Input to Primary Column Bed 3 Flow	x4:Input to Primary Column Bed 2 Flow	x5:Primary Column Feed Flow from Feed Column	x6:Primary Column Make Flow	x7:Prima Colun Bas Lev
2015-12-05 00:00:00	329.358	47.0331	2104.92	1315.580	99.0892	96.2604	54.68
2015-12-05 01:00:00	328.920	47.0293	2097.57	1539.210	98.7808	96.1749	54.83
2015-12-05 02:00:00	328.865	47.0067	2097.19	958.648	98.8306	96.4169	54.68
2015-12-05 03:00:00	329.147	47.0218	2094.45	2041.020	98.9853	96.0839	54.67
2015-12-05 04:00:00	329.256	47.0370	2096.78	1984.860	98.4337	96.1642	54.63

Print the dimensions of the new dataset.

In [5]:
```
print(df_oneweek.shape)
```

(192, 12)

2. Removing and Dropping

Create a version of the Dow dataset that does not contain any null or non-numeric values.
Print the first 5 rows of the new dataset.

In [6]:
```
def is_real_and_finite(x):
    if not np.isreal(x):
        return False
    elif not np.isfinite(x):
        return False
    elif pd.isnull(x):
        return False
    else:
        return True
```

In [7]:
```
numeric_map = df[df.columns[1:]].applymap(is_real_and_finite)
real_rows = numeric_map.all(axis = 1).values
df_dropped = df[real_rows]
df_dropped.head(5)
```

Out[7]:

	Date	x1:Primary Column Reflux Flow	x2:Primary Column Tails Flow	x3:Input to Primary Column Bed 3 Flow	x4:Input to Primary Column Bed 2 Flow	x5:Primary Column Feed Flow from Feed Column	x6:Primary Column Make Flow	x7:Pri Col
0	2015-12-01 00:00:00	327.813	45.7920	2095.06	2156.01	98.5005	95.4674	54.
1	2015-12-01 01:00:00	322.970	46.1643	2101.00	2182.90	98.0014	94.9673	54.
2	2015-12-01 02:00:00	319.674	45.9927	2102.96	2151.39	98.8229	96.0785	54.
3	2015-12-01 03:00:00	327.223	46.0960	2101.37	2172.14	98.7733	96.1223	54.
4	2015-12-01 04:00:00	331.177	45.8493	2114.06	2157.77	99.3231	94.7521	54.

5 rows × 46 columns

Drop the Avg_Delta_Composition Primary Column variable.

In [8]:
```
df_dropped_col = df_dropped.drop('Avg_Delta_Composition Primary Column', axis = 1)
df_dropped_col.head(5)
```

Out[8]:

	Date	x1:Primary Column Reflux Flow	x2:Primary Column Tails Flow	x3:Input to Primary Column Bed 3 Flow	x4:Input to Primary Column Bed 2 Flow	x5:Primary Column Feed Flow from Feed Column	x6:Primary Column Make Flow	x7:Prir Col
0	2015-12-01 00:00:00	327.813	45.7920	2095.06	2156.01	98.5005	95.4674	54.
1	2015-12-01 01:00:00	322.970	46.1643	2101.00	2182.90	98.0014	94.9673	54.
2	2015-12-01 02:00:00	319.674	45.9927	2102.96	2151.39	98.8229	96.0785	54.
3	2015-12-01 03:00:00	327.223	46.0960	2101.37	2172.14	98.7733	96.1223	54.
4	2015-12-01 04:00:00	331.177	45.8493	2114.06	2157.77	99.3231	94.7521	54.

5 rows × 45 columns

Print the observations that have any null or non-numeric values.
You may want to use `pd.isnull()` function to check whether there are any null or non-numeric values.
The expected result is just an empty `DataFrame`.

In [9]:
```
df_dropped_col[df_dropped_col.isnull().any(axis = 1)]
```

Out[9]:

	Date	x1:Primary Column Reflux Flow	x2:Primary Column Tails Flow	x3:Input to Primary Column Bed 3 Flow	x4:Input to Primary Column Bed 2 Flow	x5:Primary Column Feed Flow from Feed Column	x6:Primary Column Make Flow	x7:Primary Column Base Level	x

0 rows × 45 columns

Report the mean of *all numerical values* of this new dataset.
The output should be a single scalar value.

In [10]:
```
np.mean(df_dropped_col[df_dropped_col.columns[1:]].values)
```

Out[10]: 207.1271177980023

Report the variance of *all numerical values* of this new dataset.
The output should be a single scalar value.

In [11]:
```
np.var(df_dropped_col[df_dropped_col.columns[1:]].values)
```

Out[11]: 346970.06549771875

3. Online Data Access

Write a function that takes the simpler version of `json` file (e.g. `ethanol_simple.json`) as an argument and counts the number of C-H bonds in a given molecule.

In [12]:
```
import json

def countCH(jsonFile):
    with open(jsonFile) as f:
        mol = json.load(f)

        bonds = mol['PC_Compounds'][0]['bonds']
        atoms = mol['PC_Compounds'][0]['atoms']

        n_bonds = len(bonds['aid1']) # get the number of bonds

        count = 0
        for i in range(n_bonds):
            if atoms['element'][bonds['aid1'][i] - 1] == 6 and atoms['element'][bonds['aid2'][i] - 1
                count += 1
            elif atoms['element'][bonds['aid1'][i] - 1] == 1 and atoms['element'][bonds['aid2'][i] -
                count += 1

    return count
```

Pass `ethanol_simple.json` to this function and report the number of C-H bonds in ethanol.
We all know that there are 5 C-H bonds in ethanol. The code block below should return 5.

In [13]:
```
count = countCH('data/ethanol_simple.json')
print(count)
```

```
5
```

Write a function that takes the *CAS number* of a compound as an argument and returns the SMILES string.
The workflow should be:

- Use the PubChem RESTful API
- Extract the record as JSON
- Extract the SMILES string from this JSON
- Return the SMILES string

In [14]:
```
def returnSMILES(cas):
    r = requests.get('https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/name/{}/record/js
    r_json = json.loads(r.text)

    smiles = r_json['PC_Compounds'][0]['props'][18]['value']['sval']
    return smiles
```

Fetch the SMILES string-representation of phenol.
The CAS number of phenol is 108-95-2 .

In [15]:
```
import requests

smilesPhenol = returnSMILES('108-95-2')
print(smilesPhenol)
```

```
C1=CC=C(C=C1)O
```