Download this page as a jupyter notebook at Lab 11-TH

ES-11: Databases

Medrano, Giovanni

R11521018

ENGR 1330 Laboratory 11 - Homework

In [1]:
```
# Preamble script block to identify host, user, and kernel
import sys
! hostname
! whoami
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
DESKTOP-6HAS1BN
desktop-6has1bn\medra
C:\Users\medra\anaconda3\python.exe
3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
```

Pandas Cheat Sheet(s)

The Pandas library is a preferred tool for data scientists to perform data manipulation and analysis, next to matplotlib for data visualization and NumPy for scientific computing in Python.

The fast, flexible, and expressive Pandas data structures are designed to make real-world data analysis significantly easier, but this might not be immediately the case for those who are just getting started with it. Exactly because there is so much functionality built into this package that the options are overwhelming.

Hence summary sheets will be useful

- A summary sheet: https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf
- A different one: http://datacamp-community-prod.s3.amazonaws.com/f04456d7-8e61-482f-9cc9-da6f7f25fc9b

In [2]:
```
import pandas
import numpy
```

Exercise 1: Reading a File into a Dataframe

Pandas has methods to read common file types, such as `csv` , `xlsx` , and `json` . Ordinary text files are also quite manageable. (We will study these more in Lesson 11)

Here are the steps to follow:

1. Download the file CSV_ReadingFile.csv to your local computer
2. Run the cell below - it connects to the file, reads it into the object `readfilecsv'
3. Print the contents of the object `readfilecsv'

In [4]:
```
# download the file (do this before running the script)
readfilecsv = pandas.read_csv('CSV_ReadingFile.csv')  # Reading a .csv file
# print the contents of readfilecsv
print(readfilecsv)
```

```
    a   b   c   d
0   0   1   2   3
1   4   5   6   7
2   8   9  10  11
3  12  13  14  15
```

In [5]:
```
# How many rows are in the data table?  more code here
print('There are',readfilecsv.shape[0], 'number of rows in the data table.')
# How many columns?
print('There are',readfilecsv.shape[1], 'number of columns in the data table.')
```

```
There are 4 number of rows in the data table.
There are 4 number of columns in the data table.
```

Exercise 2

Now that you have downloaded and read a file, lets do it again, but with feeling!

Download the file named concreteData.xls to your local computer.

> The file is an Excel 97-2004 Workbook; you probably cannot inspect it within Anaconda (but maybe yes). File size is about 130K, we are going to rely on Pandas to work here!

Read the file into a dataframe object named **'concreteData'** the method name is

> - object_name = pandas.read_excel(filename)
> - It should work as above if you replace the correct placeholders

Then perform the following activities.

1. Read the file into an object

In [7]:
```
# code here looks like object_name = pandas.read_excel(filename)
data = pandas.read_excel('concreteData.xls')
print(data)
```

```
      Cement (component 1)(kg in a m^3 mixture)  \
0                                         540.0
1                                         540.0
2                                         332.5
3                                         332.5
4                                         198.6
...                                         ...
1025                                      276.4
1026                                      322.2
```

```
1027                                        148.5
1028                                        159.1
1029                                        260.9

      Blast Furnace Slag (component 2)(kg in a m^3 mixture)  \
0                                                     0.0
1                                                     0.0
2                                                   142.5
3                                                   142.5
4                                                   132.4
...                                                   ...
1025                                                116.0
1026                                                  0.0
1027                                                139.4
1028                                                186.7
1029                                                100.5

      Fly Ash (component 3)(kg in a m^3 mixture)  \
0                                              0.0
1                                              0.0
2                                              0.0
3                                              0.0
4                                              0.0
...                                            ...
1025                                          90.3
1026                                         115.6
1027                                         108.6
1028                                           0.0
1029                                          78.3

      Water  (component 4)(kg in a m^3 mixture)  \
0                                         162.0
1                                         162.0
2                                         228.0
3                                         228.0
4                                         192.0
...                                         ...
1025                                      179.6
1026                                      196.0
1027                                      192.7
1028                                      175.6
1029                                      200.6

      Superplasticizer (component 5)(kg in a m^3 mixture)  \
0                                                     2.5
1                                                     2.5
2                                                     0.0
3                                                     0.0
4                                                     0.0
...                                                   ...
1025                                                  8.9
1026                                                 10.4
1027                                                  6.1
1028                                                 11.3
1029                                                  8.6

      Coarse Aggregate  (component 6)(kg in a m^3 mixture)  \
0                                                  1040.0
1                                                  1055.0
2                                                   932.0
3                                                   932.0
4                                                   978.4
...                                                   ...
1025                                                870.1
1026                                                817.9
```

```
1027                                              892.4
1028                                              989.6
1029                                              864.5

      Fine Aggregate (component 7)(kg in a m^3 mixture)  Age (day)  \
0                                                 676.0         28
1                                                 676.0         28
2                                                 594.0        270
3                                                 594.0        365
4                                                 825.5        360
...                                                 ...        ...
1025                                              768.3         28
1026                                              813.4         28
1027                                              780.0         28
1028                                              788.9         28
1029                                              761.5         28

      Concrete compressive strength(MPa, megapascals)
0                                             79.986111
1                                             61.887366
2                                             40.269535
3                                             41.052780
4                                             44.296075
...                                                 ...
1025                                          44.284354
1026                                          31.178794
1027                                          23.696601
1028                                          32.768036
1029                                          32.401235

[1030 rows x 9 columns]
```

1. Examine the first few rows of the dataframe and describe the structure (using words) in a markdown cell just after you run the descriptor method

In [9]:
```
# code here looks like object_name.head()
data.head()
```

Out[9]:

	Cement (component 1)(kg in a m^3 mixture)	Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Fly Ash (component 3)(kg in a m^3 mixture)	Water (component 4)(kg in a m^3 mixture)	Superplasticizer (component 5)(kg in a m^3 mixture)	Coarse Aggregate (component 6)(kg in a m^3 mixture)	Fine Aggregate (component 7)(kg in a m^3 mixture)	Ag (day
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	2
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	2
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	36
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	36

1. Simplify the column names to "Cement", "BlastFurnaceSlag", "FlyAsh", "Water", "Superplasticizer", "CoarseAggregate", "FineAggregate", "Age", "CC_Strength"

In [10]:
```
# code here
names = ["Cement", "BlastFurnaceSlag", "FlyAsh", "Water", "Superplasticizer",
                  "CoarseAggregate", "FineAggregate", "Age", "CC_Strength"]
currNames = list(data.columns)

x = {}
for i, name in enumerate(currNames):
    x[name] = names[i]

data = data.rename(columns=x)

data.head()
```

Out[10]:

	Cement	BlastFurnaceSlag	FlyAsh	Water	Superplasticizer	CoarseAggregate	FineAggregate	Age	C
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	

1. Determine and report summary statistics for each of the columns.

In [12]:
```
# code here
data.describe()
```

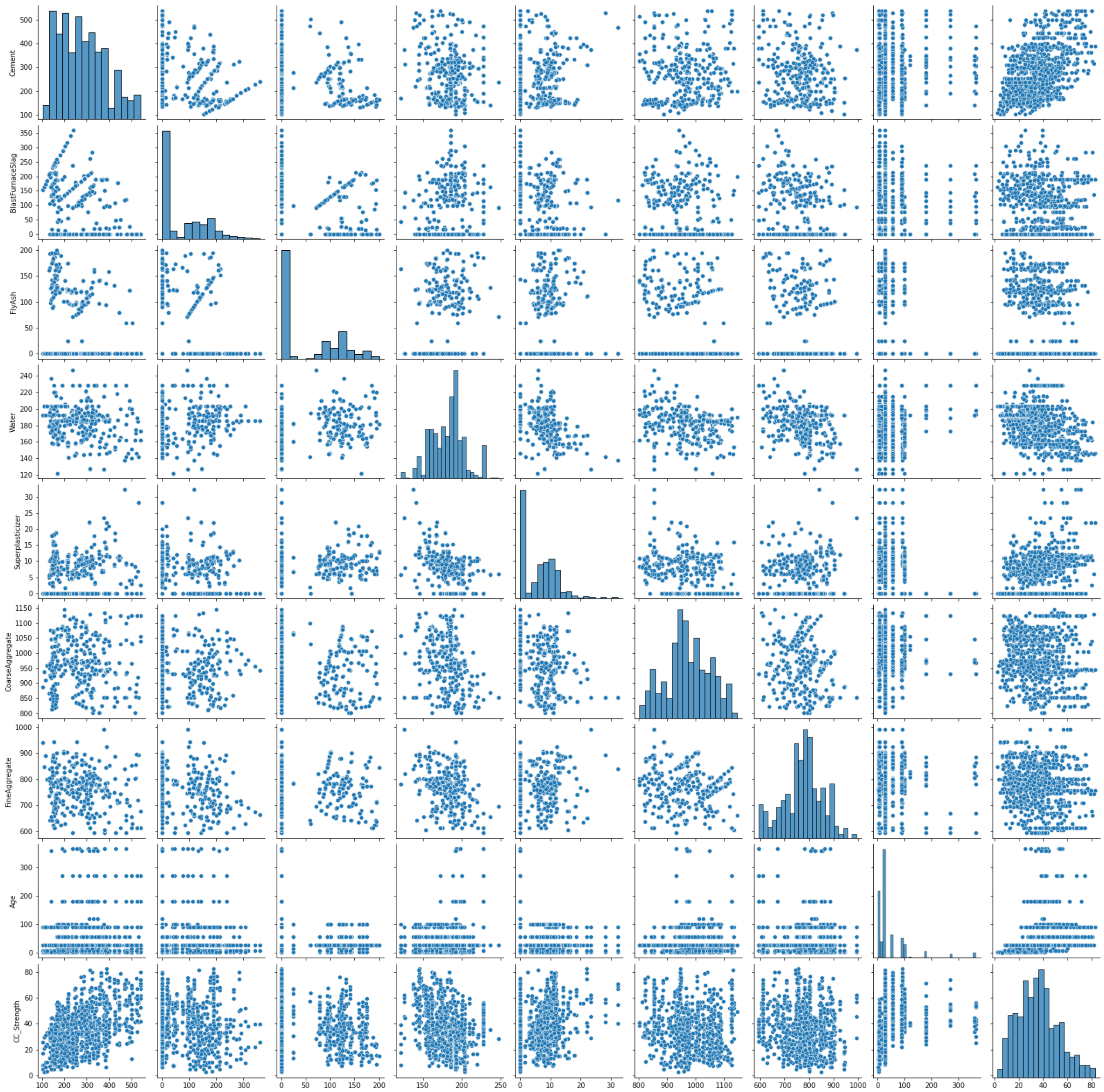
Out[12]:

	Cement	BlastFurnaceSlag	FlyAsh	Water	Superplasticizer	CoarseAggregate	Fine
count	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1
mean	281.165631	73.895485	54.187136	181.566359	6.203112	972.918592	
std	104.507142	86.279104	63.996469	21.355567	5.973492	77.753818	
min	102.000000	0.000000	0.000000	121.750000	0.000000	801.000000	
25%	192.375000	0.000000	0.000000	164.900000	0.000000	932.000000	
50%	272.900000	22.000000	0.000000	185.000000	6.350000	968.000000	
75%	350.000000	142.950000	118.270000	192.000000	10.160000	1029.400000	
max	540.000000	359.400000	200.100000	247.000000	32.200000	1145.000000	

1. Then run the script below into your notebook (after the summary statistics), describe the output (using words) in a markdown cell.

In [14]:
```
# After concreteData exists, and is non-empty; how do you know?
# then run the code block below -- It takes awhile to render output, give it a minute:
import matplotlib.pyplot
import seaborn
%matplotlib inline
```

```
seaborn.pairplot(data)
matplotlib.pyplot.show()
```



In [ ]:

```
# specify/summarize the output here!
# By utilizing all of the graphs together as a whole we can see that the data proves th
```