Hands-on Activity 6.1 Introduction to Data Analysis and Tools

CPE311 Computational Thinking with Python

Name: Pascual, Ken Leonard

Section: CPE22S3

Performed on: 04/05/2025

Submitted on: 04/05/2025

Submitted to: Engr. Roman M. Richard

6.1 Intended Learning Outcome

1. Use pandas and numpy data analysis tools

2. Demonstrate how to analyze data using numpy and pandas

6.2 Resources

- Personal Computer
- Jupyter Notebook
- Internet Connection

6.3 Supplementary Activities

Exercise 1

Run the given code below for exercises 1 and 2, perform the given tasks without usig any Python modules.

In [4]:
```
import random
random.seed(0)
salaries = [round(random.random()*1000000, -3) for _ in range(100)]
```

In [5]:
```
print(salaries)
```

```
[844000.0, 758000.0, 421000.0, 259000.0, 511000.0, 405000.0, 784000.0, 303000.0,
477000.0, 583000.0, 908000.0, 505000.0, 282000.0, 756000.0, 618000.0, 251000.0, 9
10000.0, 983000.0, 810000.0, 902000.0, 310000.0, 730000.0, 899000.0, 684000.0, 47
2000.0, 101000.0, 434000.0, 611000.0, 913000.0, 967000.0, 477000.0, 865000.0, 260
000.0, 805000.0, 549000.0, 14000.0, 720000.0, 399000.0, 825000.0, 668000.0, 1000.
0, 494000.0, 868000.0, 244000.0, 325000.0, 870000.0, 191000.0, 568000.0, 239000.
0, 968000.0, 803000.0, 448000.0, 80000.0, 320000.0, 508000.0, 933000.0, 109000.0,
551000.0, 707000.0, 547000.0, 814000.0, 540000.0, 964000.0, 603000.0, 588000.0, 4
45000.0, 596000.0, 385000.0, 576000.0, 290000.0, 189000.0, 187000.0, 613000.0, 65
7000.0, 477000.0, 90000.0, 758000.0, 877000.0, 923000.0, 842000.0, 898000.0, 9230
00.0, 541000.0, 391000.0, 705000.0, 276000.0, 812000.0, 849000.0, 895000.0, 59000
0.0, 950000.0, 580000.0, 451000.0, 660000.0, 996000.0, 917000.0, 793000.0, 82000.
0, 613000.0, 486000.0]
```

Using the data generated above, calculate the following statistics without importing anything from the statistics module i the standard library. (https://docs.python.org/3/library/statistics.html) and then confirm your results match up to those that are obtained when using the statistics module (where possible):

- Mean

In [ ]:

- Median

In [ ]:

- Mode (hint: check out the Counter in the collections module of the standard library at https://docs.python.org/3/library/collections.html#collections.Counter)

In [ ]:

- Sample variance

In [ ]:

- Sample standard deviation

In [ ]:

Exercise 2

Using the same data, calculate the following statistics using the functions in the statistics module where appropriate:

- Range

In [ ]:

- Coefficient of variation interquartile range

In [ ]:

- Quartile coefficient of dispersion

In [ ]:

Exercise 3: Pandas for Data Analysis

Load the diabetes.csv file. Convert the diabetes.csv file into dataframe.

In [10]:
```
import pandas as pd
import numpy as np
diabetes = pd.read_csv('Datasets/diabetes.csv')
```

Perform the following tasks in the diabetes dataframe:

1. Identify the column name

In [11]:
```
diabetes.columns
```

Out[11]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

2. Identify the data types of the data.

In [12]:
```
diabetes.dtypes
```

Out[12]:
```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

3. Display the total number of records.

In [13]:
```
diabetes.count
```

Out[13]:
```
<bound method DataFrame.count of      Pregnancies  Glucose  BloodPressure  Skin
Thickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]>
```

4. Display the first 20 records

In [14]:
```
diabetes.head(20)
```

Out[14]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
5	5	116	74	0	0	25.6	
6	3	78	50	32	88	31.0	
7	10	115	0	0	0	35.3	
8	2	197	70	45	543	30.5	
9	8	125	96	0	0	0.0	
10	4	110	92	0	0	37.6	
11	10	168	74	0	0	38.0	
12	10	139	80	0	0	27.1	
13	1	189	60	23	846	30.1	
14	5	166	72	19	175	25.8	
15	7	100	0	0	0	30.0	
16	0	118	84	47	230	45.8	
17	7	107	74	0	0	29.6	
18	1	103	30	38	83	43.3	
19	1	115	70	30	96	34.6	

5. Display the last 20 records

In [15]:
```
diabetes.tail(20)
```

Out[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigre
748	3	187	70	22	200	36.4	
749	6	162	62	0	0	24.3	
750	4	136	70	0	0	31.2	
751	1	121	78	39	74	39.0	
752	3	108	62	24	0	26.0	
753	0	181	88	44	510	43.3	
754	8	154	78	32	0	32.4	
755	1	128	88	39	110	36.5	
756	7	137	90	41	0	32.0	
757	0	123	72	0	0	36.3	
758	1	106	76	0	0	37.5	
759	6	190	92	0	0	35.5	
760	2	88	58	26	16	28.4	
761	9	170	74	31	0	44.0	
762	9	89	62	0	0	22.5	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

6. Change the Outcome column to Diagnosis.

In [ ]:

7. Create a new column Classification that display "Diabetes" if the value of outcome is 1, otherwise "No Diabetes".

In [ ]:

8. Create a new dataframe "withDiabetes" that gathers data with diabetes

In [ ]:

9. Create a new dataframe "noDiabetes" that gathers data with no diabetes

In [ ]:

10. Create a new dataframe "Pedia" that gathers data with age 0 to 1

In [ ]:

11. Create a new dataframe "Adult" that gathers data with age greater than 19

In [ ]:

12. Use numpy to get the average age and glucose value.

In [ ]:

13. Use numpy to get the median age and glucose value.

In [ ]:

14. Use numpy to get the middle values of glucose and age.

In [ ]:

15. Use numpy to get the standard deviation of the skinthickness.

In [ ]:

6.4 Conclusion

In [ ]: