Hands-on Activity 6.1 Introduction to Data Analysis and Tools

CPE311 Computational Thinking with Python

Name:Patrico, John Aldrin C.
Section: CPE22S3

Performed on: 04/05/2025 Submitted on: 04/05/2025

Submitted to: Engr. Roman M. Richard

6.1 Intended Learning Outcome

. Use pandas and numpy data analysis tools.

. 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 using any Python modules.

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

Using the data generated above, calculate the following statistics without importing anything from the statistics module in 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
- Median
- 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)
- Sample variance
- Sample standard deviation

In [ ]:
```
# Write a comment per statistical function
```

Exercise 2

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

- Range
- Coefficient of variation Interquartile range
- Quartile coefficient of dispersion

In [ ]:
```
# Write a comment per statistical function
```

Exercise 3: Pandas for Data Analysis

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

Perform the following tasks in the diabetes dataframe:

. Identify the column names
. Identify the data types of the data
. Display the total number of records
. Display the first 20 records
. Display the last 20 records
. Change the Outcome column to Diagnosis

. Create a new column Classification that display "Diabetes" if the value of outcome is 1 , otherwise "No Diabetes"
. Create a new dataframe "withDiabetes" that gathers data with diabetes
. Create a new dataframe "noDiabetes" thats gathers data with no diabetes
. Create a new dataframe "Pedia" that gathers data with age 0 to 19
. Create a new dataframe "Adult" that gathers data with age greater than 19
. Use numpy to get the average age and glucose value.
. Use numpy to get the median age and glucose value.
. Use numpy to get the middle values of glucose and age.
. Use numpy to get the standard deviation of the skinthickness.

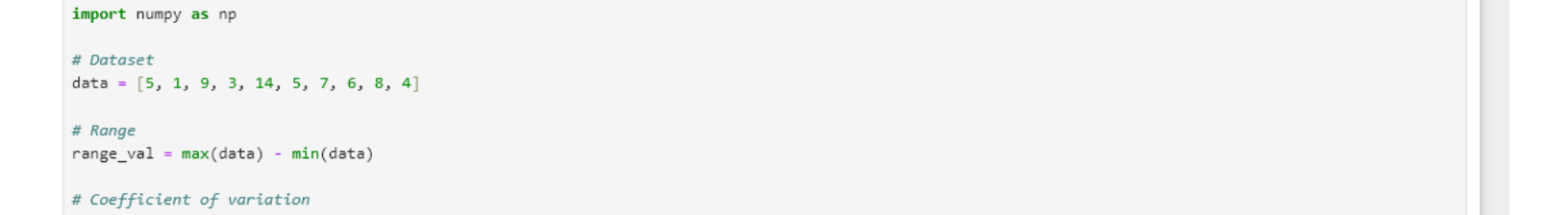
In [1]:
```
# Indicate which item you're answering with a comment
```

6.4 Conclusion

Edit this markdown block.

End.

Answer:



[1]:
```
# Dataset used for calculation
data = [5, 1, 9, 3, 14, 5, 7, 6, 8, 4]

# Mean
mean = sum(data) / len(data)

# Median
sorted_data = sorted(data)
n = len(sorted_data)
median = (sorted_data[n//2] + sorted_data[(n-1)//2]) / 2

# Mode using Counter
from collections import Counter
freq = Counter(data)
mode = [k for k, v in freq.items() if v == max(freq.values())]

# Sample Variance
mean_val = mean
variance = sum((x - mean_val) ** 2 for x in data) / (len(data) - 1)

# Sample Standard Deviation
std_dev = variance ** 0.5

print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")
print(f"Sample Variance: {variance}")
print(f"Sample Std Dev: {std_dev}")
```

```
Mean: 6.2
Median: 5.5
Mode: [5]
Sample Variance: 13.066666666666666
Sample Std Dev: 3.6147844564602556
```

[3]:
```
import statistics as stats
import numpy as np

# Dataset
data = [5, 1, 9, 3, 14, 5, 7, 6, 8, 4]

# Range
range_val = max(data) - min(data)

# Coefficient of variation
cv = stats.stdev(data) / stats.mean(data)

# Interquartile Range (IQR)
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1

# Quartile Coefficient of Dispersion
qcd = (q3 - q1) / (q3 + q1)

print(f"Range: {range_val}")
print(f"Coefficient of Variation: {cv}")
print(f"Interquartile Range: {iqr}")
print(f"Quartile Coefficient of Dispersion: {qcd}")
```

```
Range: 13
Coefficient of Variation: 0.5830297510419767
Interquartile Range: 3.5
Quartile Coefficient of Dispersion: 0.2916666666666667
```

[7]:
```
import pandas as pd
import numpy as np

# Load the diabetes.csv file
df = pd.read_csv("diabetes.csv")
```

[9]:
```
df.columns
```

[9]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

[11]:
```
df.dtypes
```

[11]:
```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

[13]:
```
len(df)
```

[13]:
```
768
```

[15]:
```
df.head(20)
```

[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1
12	10	139	80	0	0	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	0	0	0	30.0	0.484	32	1
16	0	118	84	47	230	45.8	0.551	31	1
17	7	107	74	0	0	29.6	0.254	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1

[17]:
```
df.tail(20)
```

[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
748	3	187	70	22	200	36.4	0.408	36	1
749	6	162	62	0	0	24.3	0.178	50	1
750	4	136	70	0	0	31.2	1.182	22	1
751	1	121	78	39	74	39.0	0.261	28	0
752	3	108	62	24	0	26.0	0.223	25	0
753	0	181	88	44	510	43.3	0.222	26	1
754	8	154	78	32	0	32.4	0.443	45	1
755	1	128	88	39	110	36.5	1.057	37	1
756	7	137	90	41	0	32.0	0.391	39	0
757	0	123	72	0	0	36.3	0.258	52	1
758	1	106	76	0	0	37.5	0.197	26	0
759	6	190	92	0	0	35.5	0.278	66	1
760	2	88	58	26	16	28.4	0.766	22	0
761	9	170	74	31	0	44.0	0.403	43	1
762	9	89	62	0	0	22.5	0.142	33	0
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

[35]:
```
df.rename(columns={"Outcome": "Diagnosis"}, inplace=True)
```

[39]:
```
df["Classification"] = df["Diagnosis"].apply(lambda x: "Diabetes" if x == 1 else "No Diabetes")
```

[41]:
```
withDiabetes = df[df["Diagnosis"] == 1]
```

[43]:
```
noDiabetes = df[df["Diagnosis"] == 0]
```

[45]:
```
pedia = df[df["Age"] <= 19]
```

[47]:
```
adult = df[df["Age"] > 19]
```

[49]:
```
avg_age = np.mean(df["Age"])
avg_glucose = np.mean(df["Glucose"])
print(f"Average Age: {avg_age}, Average Glucose: {avg_glucose}")
```

```
Average Age: 33.240885416666664, Average Glucose: 120.89453125
```

[51]:
```
med_age = np.median(df["Age"])
med_glucose = np.median(df["Glucose"])
print(f"Median Age: {med_age}, Median Glucose: {med_glucose}")
```

```
Median Age: 29.0, Median Glucose: 117.0
```

[53]:
```
mid_age = df["Age"].sort_values().iloc[len(df)//2]
mid_glucose = df["Glucose"].sort_values().iloc[len(df)//2]
print(f"Middle Age Value: {mid_age}, Middle Glucose Value: {mid_glucose}")
```

```
Middle Age Value: 29, Middle Glucose Value: 117
```

[55]:
```
std_skin = np.std(df["SkinThickness"], ddof=1)
print(f"Standard Deviation of SkinThickness: {std_skin}")
```

```
Standard Deviation of SkinThickness: 15.952217567727677
```

[ ]:
```
Conclusion
```

[ ]:
```
This practical exercise taught me how to use the collections module and fundamental Python functions
to manually calculate and validate important statistical statistics like mean, median, mode, variance,
and standard deviation.For further understanding,I also used the `statistics` module to calculate
the quartile coefficient of dispersion, range, interquartile range, and coefficient
of variation. In the latter section, I loaded, inspected, and analyzed the `diabetes.csv` dataset
using the pandas and numpy packages. I segmented the data into pediatric and adult groups,
renamed columns, filtered the data, and classified the data.  I determined
the average, median, and standard deviation values using Numpy, which helped me comprehend the
dataset's health indicators better.I was able to connect theoretical statistical ideas with
practical Python data analysis techniques
thanks to this exercise.
```