جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Intelligent Medical Systems Department

محاضرة العملي بلغة البايثون للمحاضرة الاولى والثانية للبيانات الاحصائية الطبية

Class: Second

Lecturer: Prof. Dr. Mehdi Ebady Manaa

Study Year: 2023-2024

Objective: to Perform a statistical study to analyze some sample data

الكود ادناه للمحاضرة الاولى

```
import numpy as np

# Step 1: Sample patient weight data (in kilograms)
weights = [65, 72, 78, 85, 90, 55, 60, 75, 80, 70]

# Step 2: Calculate basic statistics
mean_weight = np.mean(weights)  # Average weight
median_weight = np.median(weights)  # Middle value
std_dev_weight = np.std(weights)  # Standard deviation
min_weight = np.min(weights)  # Minimum weight
max_weight = np.max(weights)  # Maximum weight

# Step 3: Print the results
print(f"Mean Weight: {mean_weight:.2f} kg")
print(f"Median Weight: {median_weight:.2f} kg")
print(f"Standard Deviation: {std_dev_weight:.2f} kg")
print(f"Minimum Weight: {min_weight} kg")
print(f"Maximum Weight: {max_weight} kg")
```

الكود ادناه للمحاضرة الاولى والثانية

```
import numpy as np

# قائمة البيانات

x = [10, 0.5, 12, 0, 90, 8, 45, 2, 44, 1, 35, 5, 513, 35, 4]


# حساب الإحصائيات الأساسية
mean1 = np.mean(x)
median1 = np.median(x)
max1 = np.max(x)
```

```
min1 = np.min(x)
std1 = np.std(x)
var1 = np.var(x)
range_x = np.ptp(x)

# طباعة الإحصائيات
print(f"The mean of x is {mean1:.2f} Kg")
print(f"The median of x is {median1:.2f} Kg")
print(f"The max of x is {max1:.2f} Kg")
print(f"The min of x is {min1:.2f} Kg")
print(f"The standard deviation of x is {std1:.2f} Kg")
print(f"The variance of x is {var1:.2f} Kg")
print(f"The range of x is {range_x:.2f} Kg")

# تطبيع البيانات
normalized_x = (np.array(x) - np.min(x)) / (np.max(x) - np.min(x))
print(f"The normalized values of x are: {np.round(normalized_x, 2)} Kg")

# حساب التوزيع التكراري
def freq_distribution(data):
    unique, counts = np.unique(data, return_counts=True)
    return dict(zip(unique, counts))

freq_dis = freq_distribution(x)
print("The frequency distribution:")
for key, val in freq_dis.items():
    print(f"Value {key} appears {val} times")

# حساب عرض الفئة (اختيار عدد الفئات باستخدام قاعدة سكوت)
num_bins = int(np.ceil(1 + 3.322 * np.log10(len(x))))  # قاعدة ستورجيس لتقدير عدد الفئات
class_width = (max1 - min1) / num_bins
print(f"The class width is {class_width:.2f} Kg")
```

الكود يقوم بحساب مجموعة من **الإحصائيات الأساسية** لمجموعة بيانات x، بالإضافة إلى **تحليل التوزيع التكراري** وحساب **عرض الفئات**

x= [4 ,35 ,513 ,5 ,35 ,1 ,44 ,2 ,45 ,8 ,90 ,0 ,12 ,0.5 ,10]

حيث ان x

- x هي قائمة تحتوي على مجموعة من الأرقام تمثل البيانات التي نريد تحليلها.

ثانيا : حساب الاحصائيات

```
mean1 = np.mean(x)       # حساب المتوسط الحسابي
median1 = np.median(x)   # حساب الوسيط
max1 = np.max(x)         # إيجاد القيمة العظمى
min1 = np.min(x)         # إيجاد القيمة الصغرى
std1 = np.std(x)         # حساب الانحراف المعياري
var1 = np.var(x)         # حساب التباين
range_x = np.ptp(x)      # حساب المدى (أكبر قيمة - أصغر قيمة)
```

هذه الدوال تحسب الخصائص الإحصائية للبيانات:

الإحصائية	التفسير
المتوسط الحسابي (Mean)	مجموع القيم مقسومًا على عددها
الوسيط (Median)	القيمة التي تقع في المنتصف بعد ترتيب البيانات
القيمة العظمى (Max)	أكبر قيمة في البيانات
القيمة الصغرى (Min)	أصغر قيمة في البيانات
الانحراف المعياري (Standard Deviation)	مدى تشتت البيانات حول المتوسط
التباين (Variance)	مربع الانحراف المعياري
المدى (Range)	الفرق بين القيمة العظمى والصغرى

ثم نقوم بطباعة هذه القيم:

```
print(f"The mean of x is {mean1:.2f} Kg")
print(f"The median of x is {median1:.2f} Kg")
print(f"The max of x is {max1:.2f} Kg")
print(f"The min of x is {min1:.2f} Kg")
print(f"The standard deviation of x is {std1:.2f} Kg")
print(f"The variance of x is {var1:.2f} Kg")
print(f"The range of x is {range_x:.2f} Kg")
```

رابعا :تقيس او تطبيع البيانات(Normalization)

يعني تحويل القيم إلى نطاق بين `0` و `1` باستخدام المعادلة:

$$\frac{x - \min(x)}{\max(x) - \min(x)} = \text{normalized value}$$

نستخدم الكود

```
normalized_x = (np.array(x) - np.min(x)) / (np.max(x) - np.min(x))

print(f"The normalized values of x are: {np.round(normalized_x, 2)} Kg")
```

- `np.array(x)` : تحويل `x` إلى مصفوفة لتطبيق العمليات الحسابية.
- `np.round(normalized_x, 2)` : تقريب القيم إلى منزلتين عشريتين.

خامسا: حساب التوزيع التكراري(Frequency Distribution)

عرفت دالة التوزيع التكراري كالاتي

```
def freq_distribution(data):
    unique, counts = np.unique(data, return_counts=True)# استخراج القيم الفريدة وتكرارها
    return dict(zip(unique, counts))# تحويلها إلى قاموس (القيمة: التكرار)
```

الشرح للدالة اعلاه هو

- `np.unique(data, return_counts=True)` : ترجع القيم الفريدة وعدد مرات ظهورها.
- `zip(unique, counts)` : يربط القيم بعدد مرات تكرارها.
- `dict(...)` : تحويل النتيجة إلى قاموس.

بعدها تم حساب class width وايضا تم طباعة النتائج

.