📅 DAY 3 OF 90 DAYS AGENTIC AI CHALLENGE (NOTES + CODE)

🔍 DESCRIPTIVE STATISTICS IN ACTION

Today, I explored one of the most fundamental and powerful concepts in data analysis: **Descriptive Statistics.** These statistics help summarize and understand the main features of a dataset.

🔍 WHAT ARE DESCRIPTIVE STATISTICS?

Descriptive statistics are **summary statistics** used to describe the **basic features** of a dataset. They help you **understand** and **visualize** the data without making any predictions.

They are divided into two main types:

Type	Includes
Measures of Central Tendency	Mean, Median, Mode
Measures of Spread	Variance, Standard Deviation, Range

🧠 KEY CONCEPTS WITH EXPLANATION

1. ✅ MEAN (AVERAGE)

The **sum of all values** divided by the **number of values**

It tells you the **central value** of your dataset

Formula:

Mean=∑xn\text{Mean} = \frac{\sum x}{n}Mean=n∑x

2. ✅ VARIANCE

It measures how **spread out** the values are from the mean.

The larger the variance, the more **diverse** the data.

Sample Variance (ddof=1):

s2=∑(x−x¯)2n−1s^2 = \frac{\sum (x - \bar{x})^2}{n - 1}s2=n−1∑(x−x¯)2

3. ✅ STANDARD DEVIATION

The **square root of variance**

It brings the spread back to the original unit

Std Dev=Variance\text{Std Dev} = \sqrt{\text{Variance}}Std Dev=Variance

4. ✅ CORRELATION

Measures the **relationship** between two variables (like Hours vs Confidence)

Values:

+1: strong positive

-1: strong negative

0: no relationship

Example in Pandas:

python

CopyEdit

```
df.corr()
```

📂 HANDS-ON IN PANDAS

Let's say your dataset looks like this:

Day	Topic	Hours	Confidence
1	Pandas Basics	2	6
2	Matplotlib	2.5	8
3	Stats	3	7

You can compute stats like this:

python

CopyEdit

```
import pandas as pd df = pd.read_csv("newLearningProgress.csv") print("Mean Confidence:", df["Confidence"].mean()) print("Standard Deviation:", df["Confidence"].std()) print("Variance:", df["Confidence"].var()) print("Correlation between Hours & Confidence:") print(df[["Hours", "Confidence"]].corr())
```

✍️ MANUAL VS. PANDAS CALCULATION

I calculated descriptive statistics both **manually** using Python formulas and **efficiently** using Pandas functions like:

```
.mean(), .var(), .std(), .corr()
```

This helped me truly understand how these metrics work under the hood.

📊 WHY DESCRIPTIVE STATISTICS MATTER IN AI

Use in AI / ML	Why It's Useful
Feature Analysis	Understand which variables to use
Outlier Detection	High variance or std = suspicious points
Model Performance Check	Analyze prediction errors
Data Cleaning & Preprocessing	Normalize/scale based on mean & std

📈 VISUALIZING LEARNING PROGRESS

To make learning more engaging and reflective, I created a **personal learning tracker** that stores my daily progress in a CSV file and visualizes key metrics using **Matplotlib**.

🛠️ CODE OVERVIEW

```
import pandas as pd
import os
import matplotlib.pyplot as plt

fileName = "newLearningProgress.csv"

# Check if file exists
if os.path.exists(fileName):
    dataSet = pd.read_csv(fileName)
else:
    dataSet = pd.DataFrame(columns=["Day", "Topic",
"Hours", "Confidence"])
```

```
# User input
day = int(input("Enter the Day (1-31): "))
topic = input("Which topic did you learn today? ")
hour = float(input("How many hours did you spend? "))
confidence = float(input("How confident do you feel
(1-10)? "))

# Append new row
dataSet.loc[len(dataSet)] = [day, topic, hour,
confidence]

# Save CSV
dataSet.to_csv(fileName, index=False)

print("\n\u2705 Log Saved Successfully!\n")

# Summary Report
print("\n\ud83d\udcca Progress Report\n")
print(f"Total Days You Logged : {len(dataSet)}")
print(f"Total Hours Studied : {dataSet['Hours'].sum():.
2f} hrs")
print(f"Average Confidence :
{dataSet['Confidence'].mean():.2f}")

# Max and Min Confidence Range (mean ± std dev)
mean_conf = dataSet['Confidence'].mean()
std_conf = dataSet['Confidence'].std()

print("\n\ud83d\udcc8 Confidence Range (1 Std Dev)\n")
print(f"Max Confidence (~mean + std) : {mean_conf +
std_conf:.2f}")
print(f"Min Confidence (~mean - std) : {mean_conf -
std_conf:.2f}")
```

📊 VISUAL INSIGHTS

```
# Bar Chart: Confidence vs. Hours
df.plot(x='Hours', y='Confidence', kind='bar')
```

```
plt.title("Daily Learning Outcome")
plt.xlabel("Hours")
plt.ylabel("Confidence")
plt.grid(True)
plt.show()

# Line Chart: Confidence Over Days
df.plot(x='Day', y='Confidence', kind='line')
plt.title("Daily Learning Outcome")
plt.xlabel("Days")
plt.ylabel("Confidence")
plt.grid(True)
plt.show()
```

📌 REFLECTION

Today I truly understood how descriptive statistics guide decisions and improve insight. Combining data collection, analysis, and visualization gave me real-world practice in turning numbers into knowledge.