Student name:Zakarya Guerinat

Student ID:

SIT225: Data Capture Technologies

Activity 7.1: Data analysis and interpretation

Data analysis is a broad term that covers a wide range of techniques that enable you to reveal any insights and relationships that may exist within raw data. As you might expect, Python lends itself readily to data analysis. Once Python has analyzed your data, you can then use your findings to make good business decisions, improve procedures, and even make informed predictions based on what you've discovered.

You have done data wrangling using Python Pandas module already in activity 5.2. In this activity, you will learn Data science statistics and linear regression models.

Hardware Required

No hardware is required.

Software Required

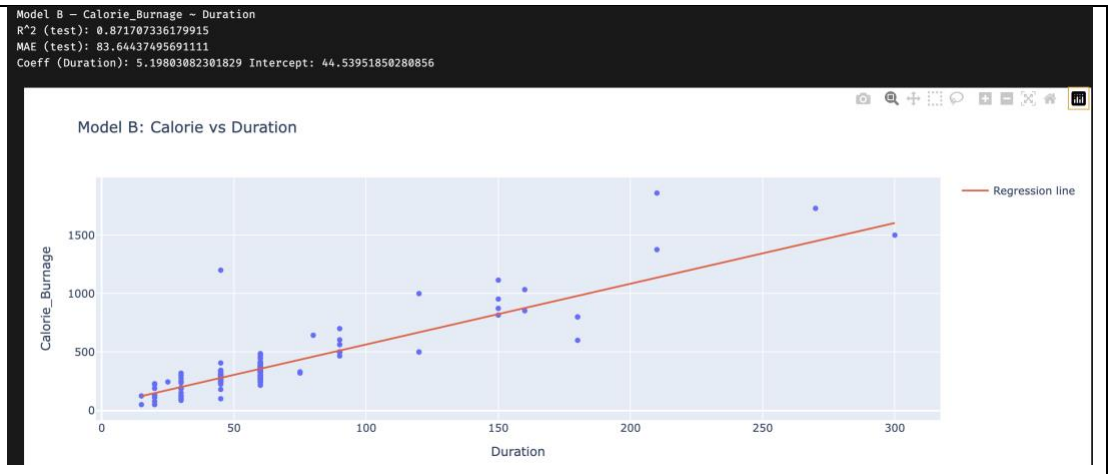
Python 3
Python packages including Pandas, Numpy, Scikit-learn, seaborn, plotly

Steps:

Step	Action
1	A Jupyter Notebook is provided for Data Science exploration here (https://github.com/deakin-deep-dreamer/sit225/tree/main/week_7). You will need to fill in your student ID and name and run all the cells to observe the output. Convert the Notebook into PDF and merge with this activity sheet which needs to be combined with this week's task for OnTrack submission.

	Question: There are sections in the Notebook. After running the cells and observing the outputs, provide your reflection in brief on the topic items for each section of the Notebook. Answer: This section summarized the dataset using count, mean, standard deviation, minimum, maximum, and percentiles. It gave me a quick understanding of the distribution of variables like Duration, Average_Pulse, Max_Pulse, and Calorie_Burnage. For example, most training sessions lasted around 60 minutes, and the average pulse was just over 100 bpm . Looking at percentiles showed how values spread out. For Average_Pulse, 25% of sessions were at or below 100 bpm, the median was around 105 bpm, and 75% were at or below 111 bpm. This helps compare distributions of Average_Pulse and Max_Pulse and see that Max_Pulse values are consistently higher but follow a similar rising trend . This part explained that even if two variables move together, it doesn't mean one causes the other. The ice cream and drowning example showed that both increase in summer due to hot weather, not because one causes the other. It reminded me to be careful when interpreting correlations in health data . Regression was used to model how Calorie_Burnage relates to other factors like Average_Pulse and Duration. The scatterplots with regression lines showed positive relationships, meaning higher pulse or longer duration usually linked with more calories burned. Comparing the two models highlighted that Duration often explained calories more directly, while Average_Pulse gave a weaker but still useful trend .
2	Question: In the 1.1 Percentile subsection of **Descriptive statistics** section in the Notebook, you have calculated 10%, 25%, 50% and 75% percentiles for *Max_Pulse*. Compare these percentiles with *Average_Pulse* percentiles for any trend, if exists. Answer: Max pulse is consistently higher across all percentiles, this makes sense because the maximum pulse for each session should be higher than the average pulse
3	Question: In the "Correlation Does not imply Causality" section answer the question regarding the increase of ice cream sale in your own understanding. Answer: This part explained that even if two variables move together, it doesn't mean one causes the other. The ice cream and drowning example showed that

	both increase in summer due to hot weather, not because one causes the other.
4	Question: In the **1.7 Linear Regression** section in the Notebook, a linear regression model was used to predict Calorie_Burnage from attributes such as Average_Pulse. The Duration value was predicted from the model for all the value range of Average_Pulse and a regression line was drawn. You will need to answer the follow up question next to 1.7 section where it is required to generate a linear regression model for Duration instead of Average_Pulse to predict the Calorie_Burnage. Take a screenshot of the regression line and paste it here. Also, comment on both the regression lines. Answer: For the first regression, the regression line is nearly flat with a very low R^2 value, meaning Average_Pulse does not strongly predict calories burned. The scatter is wide and the line does not capture much of the trend, suggesting a weak relationship. For the second regression, the regression line shows a clear positive slope, and the R^2 value is much higher (≈ 0.87). This indicates a strong relationship between workout duration and calories burned: longer sessions tend to result in higher calorie expenditure. The line follows the data pattern more closely. Duration is a much stronger predictor of Calorie_Burnage than Average_Pulse. While Average_Pulse provides some information about workout intensity, it alone does not explain calorie burn as well as the total time spent exercising.

https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=7b81716c-9638-44f8-94e7-b34b00a3eab4

In [2]:
```
# 📦 Libraries
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn.linear_model import LinearRegression

# 📂 Load CSV
df = pd.read_csv("dht22_cloud_log.csv")

# 👀 Quick look
print("Columns:", df.columns.tolist())
print("Shape:", df.shape)
df.head()
```

```
Columns: ['timestamp', 'temperature', 'humidity']
Shape: (170, 3)
```

Out[2]:

	timestamp	temperature	humidity
0	2025-08-11 16:56:47	23.700001	58.500000
1	2025-08-11 16:56:48	23.700001	58.000000
2	2025-08-11 16:56:58	23.700001	57.700001
3	2025-08-11 16:57:03	23.700001	60.099998
4	2025-08-11 16:57:08	23.700001	58.099998

In [3]:
```
# 🎯 Features (X) and target (y)
X = df[["temperature"]].values   # independent variable
y = df["humidity"].values        # dependent variable

# 🔨 Fit linear regression model
model = LinearRegression().fit(X, y)

# 📊 Coefficients
slope = model.coef_[0]
intercept = model.intercept_
r2 = model.score(X, y)

print(f"Equation: humidity = {slope:.4f} * temperature + {intercept:.4f}"
print(f"R^2: {r2:.4f}")
```

```
Equation: humidity = -2.6074 * temperature + 119.3904
R^2: 0.4201
```

In [4]:
```
# 🌡️ min and max observed temperatures
t_min, t_max = df["temperature"].min(), df["temperature"].max()
print(f"Min temp: {t_min:.2f}, Max temp: {t_max:.2f}")

# 🔮 100 equally spaced test temperatures
temp_test = np.linspace(t_min, t_max, 100).reshape(-1,1)

# 🤖 predict humidity for these temps
humidity_pred = model.predict(temp_test)
```

```
Min temp: 22.80, Max temp: 23.70
```

In [5]:
```
fig = px.scatter(df, x="temperature", y="humidity",
                 title="Temperature vs Humidity (DHT22 data)",
                 labels={"temperature":"Temperature (°C)", "humidity":"Hu

# add regression line
fig.add_scatter(x=temp_test.ravel(), y=humidity_pred, mode="lines", name=

fig.show()
```

In [6]:
```
# 🧹 filter out possible outliers
df_filtered = df[(df["temperature"] > 22.85) & (df["temperature"] < 23.65
print("Before:", df.shape, "After filtering:", df_filtered.shape)

# retrain model
Xf = df_filtered[["temperature"]].values
yf = df_filtered["humidity"].values
model_f = LinearRegression().fit(Xf, yf)

slope_f = model_f.coef_[0]
intercept_f = model_f.intercept_
r2_f = model_f.score(Xf, yf)

print(f"Filtered Equation: humidity = {slope_f:.4f}*temperature + {interc
print(f"Filtered R^2: {r2_f:.4f}")

# predictions for trend line
temp_test_f = np.linspace(df_filtered["temperature"].min(),
                          df_filtered["temperature"].max(), 100).reshape(
humidity_pred_f = model_f.predict(temp_test_f)

# plot
fig = px.scatter(df_filtered, x="temperature", y="humidity",
                 title="Filtered Temperature vs Humidity",
                 labels={"temperature":"Temperature (°C)", "humidity":"Hu
fig.add_scatter(x=temp_test_f.ravel(), y=humidity_pred_f,
                mode="lines", name="Regression line (filtered)")
fig.show()
```

```
Before: (170, 3) After filtering: (145, 3)
Filtered Equation: humidity = -2.9894*temperature + 128.1277
Filtered R^2: 0.3830
```

In [7]:
```
# 🧹 filter to remove more extremes: keep humidity only between 58% and 6
df_filtered2 = df[(df["humidity"] >= 58) & (df["humidity"] <= 60)]
print("Before:", df.shape, "After filtering:", df_filtered2.shape)

# retrain model
X2 = df_filtered2[["temperature"]].values
y2 = df_filtered2["humidity"].values
model2 = LinearRegression().fit(X2, y2)

slope2 = model2.coef_[0]
intercept2 = model2.intercept_
r2_2 = model2.score(X2, y2)

print(f"Tighter Filter Equation: humidity = {slope2:.4f}*temperature + {i
print(f"Tighter Filter R^2: {r2_2:.4f}")

# predictions
```

```
temp_test2 = np.linspace(df_filtered2["temperature"].min(),
                         df_filtered2["temperature"].max(), 100).reshape(
humidity_pred2 = model2.predict(temp_test2)

# plot
fig = px.scatter(df_filtered2, x="temperature", y="humidity",
                 title="Temperature vs Humidity (tighter filtered)",
                 labels={"temperature":"Temperature (°C)", "humidity":"Hu
fig.add_scatter(x=temp_test2.ravel(), y=humidity_pred2,
                mode="lines", name="Regression line (tighter filter)")
fig.show()
```

```
Before: (170, 3) After filtering: (101, 3)
Tighter Filter Equation: humidity = -1.2659*temperature + 88.2723
Tighter Filter R^2: 0.1847
```

QUESTION 2

Scenario 1 – Original model (all 170 samples) • Equation: humidity = −2.61 × temperature + 119.39 • $R^2 \approx 0.42$ • Interpretation: • The regression line slopes downward, showing a negative relationship: as temperature increases, humidity decreases. • The line follows the general cloud of points but does not explain all the variation (only ~42%). • Outliers are clearly present (some humidity points above 61% or below 57%), which influence the slope and intercept. • The model suggests a moderate linear trend but with a fair amount of scatter.

Scenario 2 – Filtered outliers (145 samples) • Equation: humidity = −2.99 × temperature + 128.13 • $R^2 \approx 0.38$ • Interpretation: • After removing 25 extreme samples (high and low temperatures), the slope got steeper (−2.99 vs −2.61). • This means that once the extremes are gone, the model finds an even stronger negative relationship between temperature and humidity. • However, R^2 decreased slightly (0.38 vs 0.42). • Why? Because we reduced the variability in the data, leaving less room for the regression line to explain variation. • The line looks “tighter” around the main cluster, but still captures the downward trend.

Scenario 3 – Tighter filtering (101 samples, humidity restricted to 58–60%) • Equation: humidity = −1.27 × temperature + 88.27 • $R^2 \approx 0.18$ • Interpretation: • The slope is now much flatter (−1.27 compared to −2.99). • By restricting to a narrow band of humidity values, the negative relationship between temperature and humidity becomes much weaker. • R^2 dropped sharply (0.18), meaning the model barely explains the variation. • Essentially, once outliers are removed and only a tight cluster remains, the linear model doesn’t have enough spread to show a strong trend. • The regression line is less meaningful here because the data doesn’t vary enough to reveal a strong linear pattern.

Comparing the Scenarios • Original: Outliers influenced the slope but provided more variance → moderate R^2 (0.42). • Filtered extremes: Stronger slope, but less variance → slightly weaker R^2 (0.38). • Tighter filter: Very flat slope, very weak R^2 (0.18) → shows that without variance, the regression model cannot capture a strong relationship.

Overall Insight • Outliers can pull the regression line and make the relationship look stronger or weaker. • Filtering reduces noise, but too much filtering can make the model meaningless because you're only looking at a very narrow range of values. • The best balance is usually removing clear outliers but keeping enough data spread so the regression captures the true trend.