**Homework Task 1: Handling Missing Values**

**Activity Goal:** Practice different techniques for dealing with missing data.

**Tasks:**

1. Identify columns with more than **20% missing values** and **drop** them.

#Hint: In first step , create a variable missing\_person and calculate it using df.isna().sum() and len(df).

# Use simply process of calculating a percentage that is missing values/total values\*100.

1. Fill missing values in the **numerical columns** using **median** instead of mean.

#Hint Use df.fillna and median commands

df.fillna(df.median(), inplace=True)

1. Fill missing values in the **categorical columns** using the **most frequent value (mode)**.

df["Column\_Name"].fillna(df["Column\_Name"].mode()[0], inplace=True)

🔹 **Expected Outcome:** Students should learn to analyze and decide the best way to handle missing values.

**Homework Task 2: Advanced Filtering and Indexing**

**Activity Goal:** Use filtering and indexing techniques to extract meaningful data subsets.

**Tasks:**

1. Retrieve all properties located in **"Richmond"** with a **price greater than $1,000,000**.

#Hint: You should use is equal == command for location “Richmond” and greater than > for price.

richmond\_houses = df[(df["Suburb"] == "Richmond") & (df["Price"] > 1000000)]

print(richmond\_houses.head())

1. Extract only the **"Price", "Suburb", and "BuildingArea"** columns for properties where the land size is above 500 square meters.

df\_filtered = df.loc[df["Landsize"] > 500, ["Price", "Suburb", "BuildingArea"]]

1. Find the **top 5 most expensive houses** in the dataset using sorting.

#Hint: You should use sort option by Price and then print the top 5 rows.

df\_sorted = df.sort\_values(by="Price", ascending=False)

print(df\_sorted.head(5))

🔹 **Expected Outcome:** Students should be able to extract, filter, and sort data efficiently.

**Homework Task 3: Data Transformation & Feature Engineering**

**Activity** **Goal:** Apply transformation techniques to create new insights from the data.

**Tasks:**

1. Create a **new column** that calculates the **price per room** (Price / Rooms).

df["Price\_per\_Room"] = df["Price"] / df["Rooms"]

1. Convert the **"Date" column** (if available) into a proper datetime format.

df["Date"] = pd.to\_datetime(df["Date"])

1. Extract the **year of sale** from the date and create a new column Year\_Sold.

df["Year\_Sold"] = df["Date"].dt.year

🔹 **Expected Outcome:** Students should learn how to generate new insights by transforming data.

**Homework Task 4: Aggregation and Grouping**

**Activity** **Goal:** Learn how to group and aggregate data for analysis.

**Tasks:**

1. Find the **average price of properties** in each suburb.

suburb\_avg\_price = df.groupby("Suburb")["Price"].mean()

print(suburb\_avg\_price)

1. Find the **total number of properties sold** in each suburb.

#Hint: For properties sold, you should use the price column because the Price represent the price at which the properties are sold

# You can use groupby option

properties\_per\_suburb = df.groupby("Suburb")["Price"].count()

print(properties\_per\_suburb)

1. Identify the **suburb with the highest average price**.

most\_expensive\_suburb = suburb\_avg\_price.idxmax()

print("Most expensive suburb:", most\_expensive\_suburb)

🔹 **Expected Outcome:** Students will understand how to summarize data using aggregation.

**📌 Homework Task 5: Exporting & Reporting Data**

✅ **Goal:** Save cleaned and processed data for further use.

**Tasks:**

1. Save the cleaned dataset into a **new CSV file**.

suburb\_avg\_price.head(10).plot(kind="bar")

plt.title("Average House Prices in Top 10 Suburbs")

plt.xlabel("Suburb")

**🔹 Bonus Challenge (For Advanced Students)**

* Use **apply()** to create a new column that classifies houses as **"Expensive"** or **"Affordable"**, based on whether the price is above or below the median price.

median\_price = df["Price"].median()

df["Category"] = df["Price"].apply(lambda x: "Expensive" if x > median\_price else "Affordable")

* Find the **correlation between price and the number of rooms** using .corr().

**💡 Summary**

| **Task** | **Concept Practiced** |
| --- | --- |
| Handling Missing Values | Drop/fill missing data |
| Filtering and Indexing | Extract specific data using conditions |
| Data Transformation | Create new columns from existing data |
| Aggregation | Group data and compute summaries |
| Exporting Data | Save cleaned data and reports |
| Bonus Challenge | Advanced Pandas operations |

**Submission Guidelines**

* Students must submit:
  + **Python script (.py) or Jupyter Notebook (.ipynb)**
  + **Processed CSV file (cleaned\_melb\_data.csv)**
  + **Summary report (summary\_report.txt)**
* You should upload the file on your git hub and the CR will compile a list of links and send it to me.
* Deadline: **Monday 24th , 10 pm.**