WEEK4

Q1.

```
import pandas as pd

data = {
    "Year": [
        2022, 2021, 2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013,
2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005,
        2004, 2003, 2002, 2001, 2000, 1999, 1998, 1997, 1996, 1995,
1994,
        1993, 1992, 1991, 1990, 1989, 1988, 1987, 1986, 1985, 1984,
1983, 1982, 1981, 1980, 1979, 1978, 1977, 1976,
        1975, 1974, 1973, 1972, 1971, 1970, 1969, 1968, 1967, 1966,
1965
    ],
    "Price_24k_10g": [
        52550, 50045, 48651, 35220, 31438, 29667, 28623, 26343, 29245,
29500, 31500, 26400, 18550, 14500, 12500, 10800,
        8400, 7000, 5850, 5660, 4990, 4300, 4400, 4234, 4045, 4275,
5160, 4680, 4598,
        4140, 4334, 3466, 3200, 3140, 3130, 2570, 2140, 2130, 1570,
1800, 1645, 1800, 1330, 937, 685, 486, 432,
        540, 412, 279, 202, 193, 184, 176, 162, 103, 94, 72
    ]
}

df = pd.DataFrame(data)

csv_path = "gold_price_data.csv"
df.to_csv(csv_path, index=False)

csv_path
```

```
'gold_price_data.csv'
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import math
import pandas as pd
import numpy as np

X = df["Year"].values.reshape(-1, 1)
y = df["Price_24k_10g"].values

model = LinearRegression()
model.fit(X, y)

sklearn_coef = model.coef_[0]
```

```
sklearn_intercept = model.intercept_

y_pred_sklearn = model.predict(X)
mse_sklearn = mean_squared_error(y, y_pred_sklearn)
rmse_sklearn = math.sqrt(mse_sklearn)

X_mean = np.mean(df["Year"])
y_mean = np.mean(df["Price_24k_10g"])
b1 = np.sum((df["Year"] - X_mean) * (df["Price_24k_10g"] - y_mean)) /
np.sum((df["Year"] - X_mean) ** 2)
b0 = y_mean - b1 * X_mean

y_pred_manual = b0 + b1 * df["Year"]
mse_manual = mean_squared_error(y, y_pred_manual)
rmse_manual = math.sqrt(mse_manual)

price_10g_2025 = model.predict([[2025]])[0]
price_1g_2025 = price_10g_2025 / 10

print("=== Simple Linear Regression Results ===")
print(f"1. Sklearn Coefficient (Slope): {sklearn_coef:.4f}")
print(f"2. Sklearn Intercept: {sklearn_intercept:.4f}")
print(f"3. Manual Coefficient (Slope): {b1:.4f}")
print(f"4. Manual Intercept: {b0:.4f}")
print(f"5. Sklearn MSE: {mse_sklearn:.2f}")
print(f"6. Sklearn RMSE: {rmse_sklearn:.2f}")
print(f"7. Manual MSE: {mse_manual:.2f}")
print(f"8. Manual RMSE: {rmse_manual:.2f}")
print(f"9. Predicted Price for 2025 (10g): ₹{price_10g_2025:.2f}")
print(f"10. Predicted Price for 2025 (1g): ₹{price_1g_2025:.2f}")

=== Simple Linear Regression Results ===
1. Sklearn Coefficient (Slope): 676.9157
2. Sklearn Intercept: -1339354.2454
3. Manual Coefficient (Slope): 676.9157
4. Manual Intercept: -1339354.2454
5. Sklearn MSE: 62748523.33
6. Sklearn RMSE: 7921.40
7. Manual MSE: 62748523.33
8. Manual RMSE: 7921.40
9. Predicted Price for 2025 (10g): ₹31400.02
10. Predicted Price for 2025 (1g): ₹3140.00
```

Q2.

```
import pandas as pd

gold_data = {
    "Year": [
        1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
```

```
1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
        2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009,
2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
        2019, 2020, 2021, 2022
    ],
    "Gold_24k_10g": [
        1800, 1645, 1800, 1330, 937, 685, 486, 432, 540, 412, 279,
202, 193, 184, 176, 162, 103, 94, 72, 4400, 4300, 4990,
        5660, 5850, 7000, 8400, 10800, 12500, 14500, 18550, 26400,
31500, 29500, 29245, 26343, 28623, 29667, 31438, 35220,
        48651, 50045, 52550
    ]
}

silver_data = {
    "Year": [
        1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
        2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009,
2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
        2019, 2020, 2021, 2022
    ],
    "Silver_Rs_per_kg": [
        2715, 2720, 3105, 3570, 3955, 4015, 4794, 6066, 6755, 6463,
6646, 8040, 5489, 7124, 6335, 7346, 7345, 8560, 7615,
        7900, 7215, 7875, 7695, 11770, 10675, 17405, 19520, 23625,
22165, 27255, 56900, 56290, 54030, 43070, 37825, 36990,
        37825, 41400, 40600, 63435, 62572, 55100
    ]
}

df_gold = pd.DataFrame(gold_data)
df_silver = pd.DataFrame(silver_data)

df = pd.merge(df_gold, df_silver, on="Year")

csv_path = "gold_silver_price_data.csv"
df.to_csv(csv_path, index=False)

csv_path
```

```
'gold_silver_price_data.csv'
```

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import math

df = pd.read_csv("gold_silver_price_data.csv")
```

```
X = df[["Year"]].values

y_gold = df["Gold_24k_10g"].values
y_silver = df["Silver_Rs_per_kg"].values

model_gold = LinearRegression()
model_gold.fit(X, y_gold)
coef_gold = model_gold.coef_[0]
intercept_gold = model_gold.intercept_
y_pred_gold = model_gold.predict(X)
mse_gold = mean_squared_error(y_gold, y_pred_gold)
rmse_gold = math.sqrt(mse_gold)

model_silver = LinearRegression()
model_silver.fit(X, y_silver)
coef_silver = model_silver.coef_[0]
intercept_silver = model_silver.intercept_
y_pred_silver = model_silver.predict(X)
mse_silver = mean_squared_error(y_silver, y_pred_silver)
rmse_silver = math.sqrt(mse_silver)

def manual_regression(x, y):
    x_mean = np.mean(x)
    y_mean = np.mean(y)
    b1 = np.sum((x - x_mean) * (y - y_mean)) / np.sum((x - x_mean) **
2)
    b0 = y_mean - b1 * x_mean
    y_pred = b0 + b1 * x
    mse = mean_squared_error(y, y_pred)
    rmse = math.sqrt(mse)
    return b1, b0, mse, rmse

b1_gold, b0_gold, mse_gold_manual, rmse_gold_manual =
manual_regression(df["Year"].values, y_gold)
b1_silver, b0_silver, mse_silver_manual, rmse_silver_manual =
manual_regression(df["Year"].values, y_silver)

gold_10g_2024 = model_gold.predict([[2024]])[0]
gold_1g_2024 = gold_10g_2024 / 10

silver_1kg_2024 = model_silver.predict([[2024]])[0]
silver_1g_2024 = silver_1kg_2024 / 1000

print("SKLEARN GOLD: Coef =", coef_gold, "Intercept =",
intercept_gold, "MSE =", mse_gold, "RMSE =", rmse_gold)
print("MANUAL GOLD:  Coef =", b1_gold, "Intercept =", b0_gold, "MSE
=", mse_gold_manual, "RMSE =", rmse_gold_manual)

print("SKLEARN SILVER: Coef =", coef_silver, "Intercept =",
```

```
intercept_silver, "MSE =", mse_silver, "RMSE =", rmse_silver)
print("MANUAL SILVER:  Coef =", b1_silver, "Intercept =", b0_silver,
"MSE =", mse_silver_manual, "RMSE =", rmse_silver_manual)

print(f"\nPredicted GOLD price for 1g in 2024: Rs.{gold_1g_2024:.2f}")
print(f"Predicted SILVER price for 1g in 2024: Rs.
{silver_1g_2024:.2f}")
```

```
SKLEARN GOLD: Coef = 1118.9529211571185 Intercept = -2227020.843124544
MSE = 57238132.02641182 RMSE = 7565.588676792561
MANUAL GOLD:  Coef = 1118.9529211571185 Intercept = -2227020.843124544
MSE = 57238132.02641182 RMSE = 7565.588676792561
SKLEARN SILVER: Coef = 1388.7457256300138 Intercept = -
2759150.8793722824 MSE = 94699881.65523116 RMSE = 9731.38641999336
MANUAL SILVER:  Coef = 1388.7457256300138 Intercept = -
2759150.8793722824 MSE = 94699881.65523116 RMSE = 9731.38641999336

Predicted GOLD price for 1g in 2024: Rs.3773.99
Predicted SILVER price for 1g in 2024: Rs.51.67
```