

PacificPlateGaussianDrift: Least Squares + Gaussian Noise

Concept (What are we estimating?)

The Hawaiian-Emperor seamount chain is like a breadcrumb trail: volcanoes form near a hotspot, then the Pacific Plate moves, carrying older volcanoes away. So **distance from the hotspot** increases roughly linearly with **age**.

We want to estimate the plate's **drift velocity** (in km/Myr) from data:

- **age** of each volcanic island/seamount (million years)
 - **distance** from a reference (km)
-

Mathematical Model (Linear + Gaussian noise)

We assume a simple physical relationship:

distance = velocity × age + noise

More explicitly, a linear model with an intercept:

$$d_i = b + v t_i + \epsilon_i$$

- d_i : distance (km)
- t_i : age (Myr)
- v : plate velocity (km/Myr) ← **our main estimate**
- b : intercept (km) (accounts for reference offset, imperfect hotspot origin, etc.)
- $\epsilon_i \sim N(0, \sigma^2)$: Gaussian noise

This is “Gaussian Drift”: the drift is linear (deterministic), but observations have Gaussian measurement / modeling error.

Least Squares (why this estimator?)

Stack all samples into matrix form:

$$y = Xw + \epsilon$$

where

$$y = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}, X = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix}, w = \begin{bmatrix} b \\ v \end{bmatrix}$$

Least Squares solves:

$$\hat{w} = \arg \min_w \|y - X w\|_2^2$$

Closed-form solution:

$$\hat{w} = (X^T X)^{-1} X^T y$$

And here's the key link:

If ϵ_i are i.i.d. Gaussian, then **Least Squares = Maximum Likelihood Estimation (MLE)** for b, v . So the probabilistic assumption justifies the LS estimator.

How we judge “how good” the estimate is

After fitting b, v , we will evaluate:

1. **Plot** distance vs age + fitted line (sanity check)
 2. **Residuals** $r_i = d_i - \hat{d}_i$
 3. Metrics:
 - RMSE
 - R^2
 4. Basic uncertainty:
 - Estimate σ^2 from residuals
 - Parameter covariance and 95% CI for v (velocity)
-

Load the dataset + sanity checks

We now load the dataset from the repo structure:

PacificPlateGaussianDrift/ data/volcanoes_data.csv
 notebooks/pacific_plate_gaussian_drift.ipynb

Since the notebook is inside `notebooks/`, we access the CSV using:

```
../data/volcanoes_data.csv
```

We will:

- load the CSV.
- inspect shape and columns.
- detect likely age and distance columns.

- preview first rows.

```
import pandas as pd
import numpy as np

CSV_PATH = "../data/volcanoes_data.csv"

df = pd.read_csv(CSV_PATH)

print("Shape:", df.shape)
print("\nColumns:", list(df.columns))

display(df.head(10))
display(df.describe())

# Try to auto-detect likely 'age' and 'distance' columns (case-
insensitive)
age_candidates = [c for c in df.columns if "age" in c.lower()]
dist_candidates = [c for c in df.columns if "dist" in c.lower() or
"distance" in c.lower()]

print("\nAge column candidates:", age_candidates)
print("Distance column candidates:", dist_candidates)

if len(age_candidates) == 1 and len(dist_candidates) == 1:
    age_col = age_candidates[0]
    dist_col = dist_candidates[0]
    print(f"\nAuto-selected: age_col='{age_col}',
dist_col='{dist_col}'")
    display(df[[age_col, dist_col]].head(10))
else:
    print("\nMultiple or unclear column matches. We'll manually select
in next part.")
```

Shape: (36, 4)

Columns: ['1', 'Kilauea', '0', '0.4']

	1	Kilauea	0	0.4
0	3	Mauna Kea	54	0.375
1	5	Kohala	100	0.430
2	6	Haleakala	182	0.750
3	7	Kahoolawe	185	1.030
4	8	West Maui	221	1.320
5	9	Lanai	226	1.280
6	10	East Molokai	256	1.760
7	11	West Molokai	280	1.900
8	12	Koolau	339	2.600
9	13	Waianae	374	3.700

		0	0.4
count		36.000000	36.000000

mean	1877.194444	21.820417
std	1588.092815	20.758495
min	54.000000	0.375000
25%	365.250000	3.425000
50%	1345.500000	12.650000
75%	3333.250000	39.000000
max	4860.000000	64.700000

Age column candidates: []
Distance column candidates: []

Multiple or unclear column matches. We'll manually select in next part.

Visualization: Age vs Distance

We now explicitly select:

- Age column → '0.4'
- Distance column → '0'

We convert them to numeric (safety step), then create a scatter plot:

distance vs age

If the Pacific Plate drift model is reasonable, we should see an approximately linear upward trend.

```
import matplotlib.pyplot as plt

# Manually selecting based on inspection
age_col = '0.4'
dist_col = '0'

# Ensure numeric
df[age_col] = pd.to_numeric(df[age_col], errors='coerce')
df[dist_col] = pd.to_numeric(df[dist_col], errors='coerce')

# Drop any rows that failed conversion (should be none)
df_clean = df[[age_col, dist_col]].dropna()

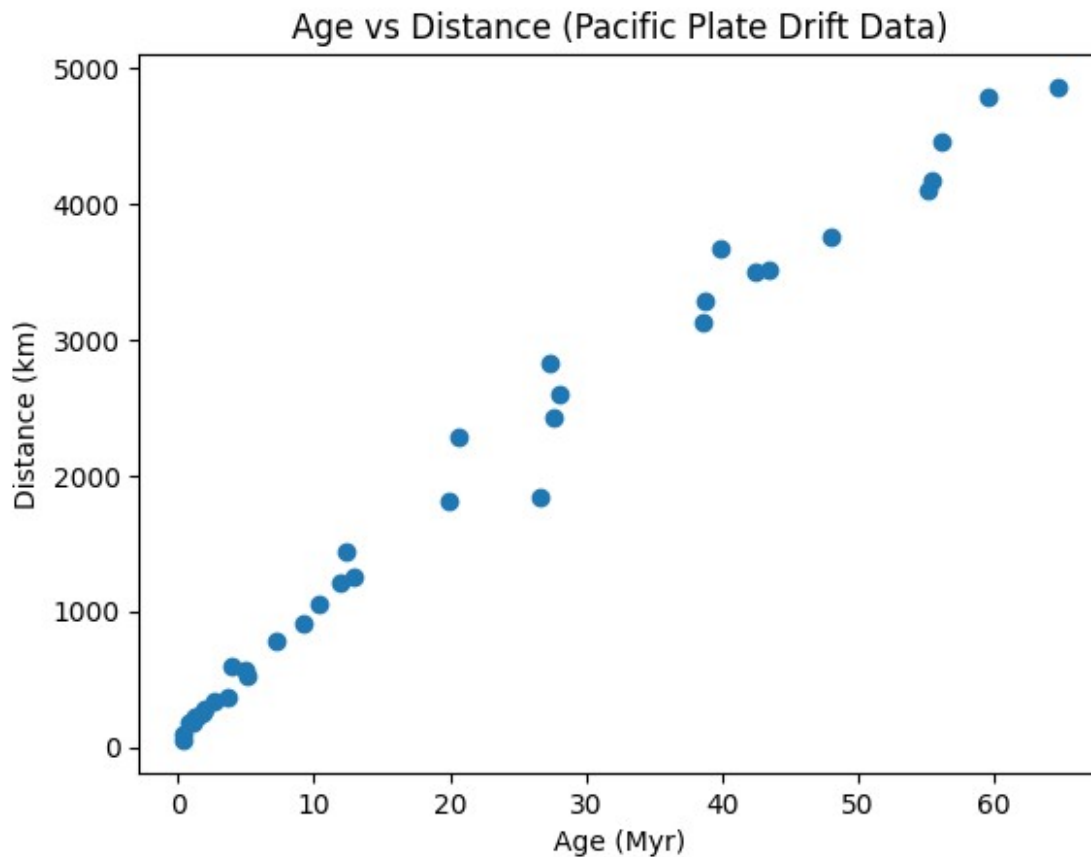
age = df_clean[age_col].values
distance = df_clean[dist_col].values

print("Final dataset size:", len(age))

plt.figure()
plt.scatter(age, distance)
plt.xlabel("Age (Myr)")
```

```
plt.ylabel("Distance (km)")
plt.title("Age vs Distance (Pacific Plate Drift Data)")
plt.show()
```

Final dataset size: 36



Closed-Form Least Squares Estimation

We now build the design matrix:

$X = [1, \text{age}]$

and solve the normal equation:

$$w_{hat} = (X^T X)^{(-1)} X^T y$$

This gives us:

- $b_{hat} \rightarrow$ intercept
- $v_{hat} \rightarrow$ Pacific Plate velocity (km/Myr)

This is the exact closed-form Least Squares solution.

```

# Build design matrix X
N = len(age)
X = np.column_stack((np.ones(N), age))
y = distance.reshape(-1, 1)

# Closed-form solution
XtX = X.T @ X
XtX_inv = np.linalg.inv(XtX)
XtY = X.T @ y

w_hat = XtX_inv @ XtY

b_hat = w_hat[0, 0]
v_hat = w_hat[1, 0]

print("Closed-form Least Squares Solution")
print()
print(f"Intercept ( $\hat{b}$ ): {b_hat:.4f} km")
print(f"Velocity ( $\hat{v}$ ): {v_hat:.4f} km/Myr")

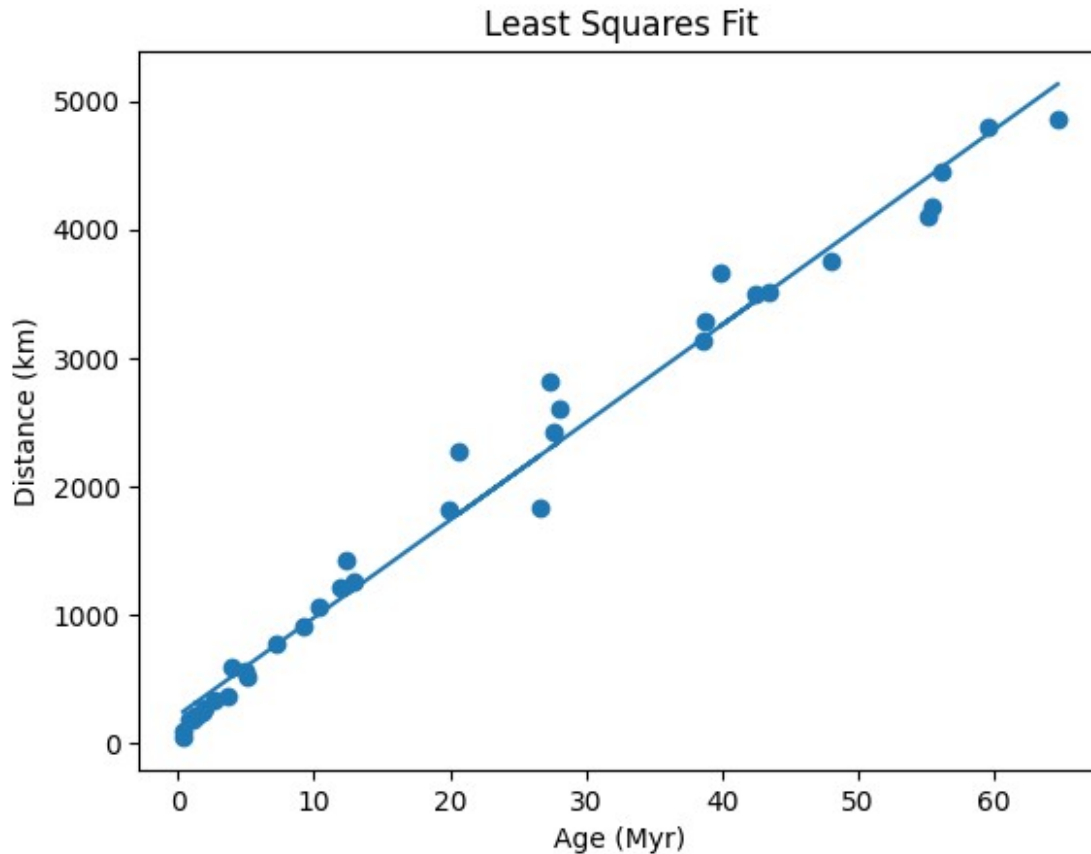
# Also compute predictions
y_pred = (X @ w_hat).flatten()

# Quick visual check
plt.figure()
plt.scatter(age, distance)
plt.plot(age, y_pred)
plt.xlabel("Age (Myr)")
plt.ylabel("Distance (km)")
plt.title("Least Squares Fit")
plt.show()

```

Closed-form Least Squares Solution

Intercept (\hat{b}): 221.6579 km
Velocity (\hat{v}): 75.8710 km/Myr



Verify with `numpy.linalg.lstsq`

We now compute the least squares solution using NumPy's built-in solver:

```
np.linalg.lstsq(X, y)
```

If our closed-form implementation is correct, the estimated parameters should match (up to numerical precision).

```
# Using numpy's least squares solver
w_np, residuals_np, rank, s = np.linalg.lstsq(X, y, rcond=None)

b_np = w_np[0, 0]
v_np = w_np[1, 0]

print("NumPy Least Squares Solution")
print()
print(f"Intercept ( $\hat{b}$ ): {b_np:.6f} km")
print(f"Velocity ( $\hat{v}$ ): {v_np:.6f} km/Myr")

print("\nDifference from closed-form:")
```

```
print(f"Intercept diff: {abs(b_np - b_hat):.10f}")
print(f"Velocity diff: {abs(v_np - v_hat):.10f}")
```

NumPy Least Squares Solution

```
Intercept ( $\hat{b}$ ): 221.657910 km
Velocity ( $\hat{v}$ ): 75.870986 km/Myr
```

```
Difference from closed-form:
Intercept diff: 0.0000000000
Velocity diff: 0.0000000000
```

Goodness of Fit

We now evaluate how good our estimate is.

We compute:

- Residuals: $r = y - y_{\hat{a}}$
- $RMSE$ (root mean squared error)
- R^2 (coefficient of determination)

We also plot:

1. Residuals vs Age
2. Histogram of residuals

If the residuals look randomly distributed around zero, our linear + Gaussian noise model is appropriate.

```
# Residuals
residuals = distance - y_pred

# RMSE
rmse = np.sqrt(np.mean(residuals**2))

# R^2
ss_total = np.sum((distance - np.mean(distance))**2)
ss_res = np.sum(residuals**2)
r2 = 1 - (ss_res / ss_total)

print("Goodness of Fit")
print()
print(f"RMSE: {rmse:.4f} km")
print(f"R^2 : {r2:.6f}")

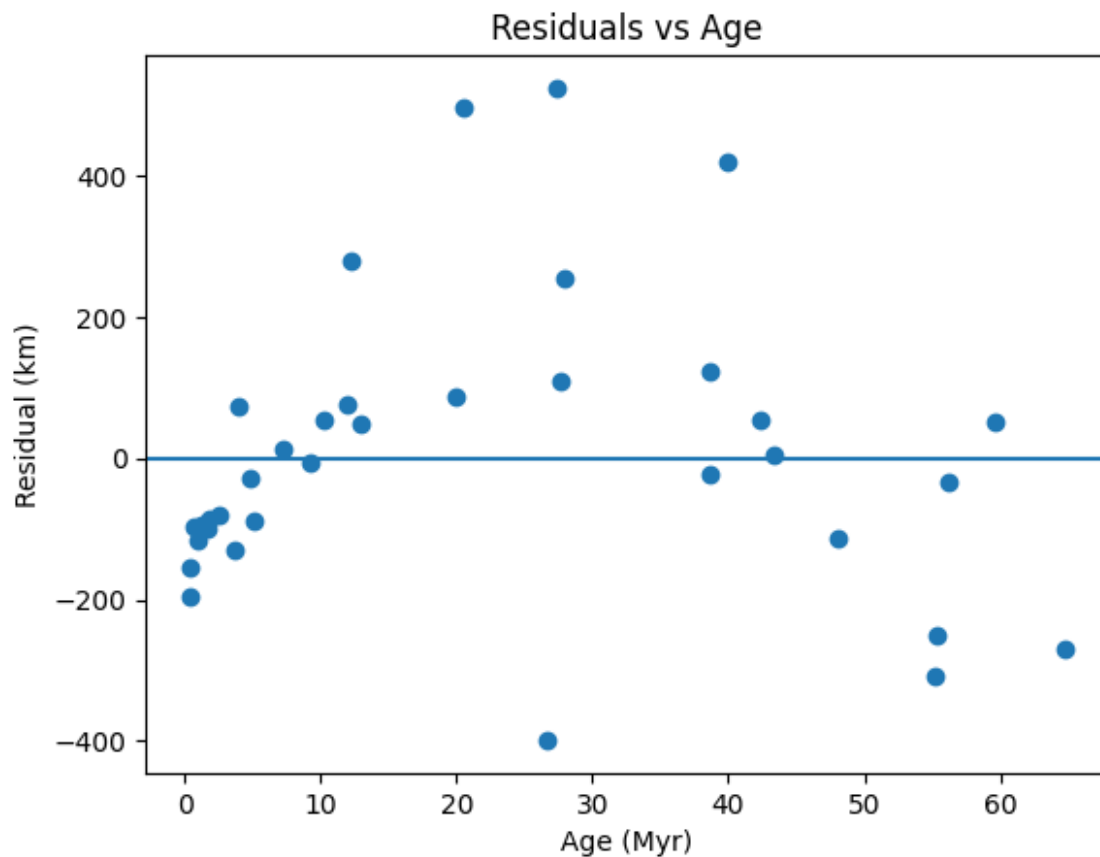
# Residual plot
plt.figure()
plt.scatter(age, residuals)
plt.axhline(0)
```

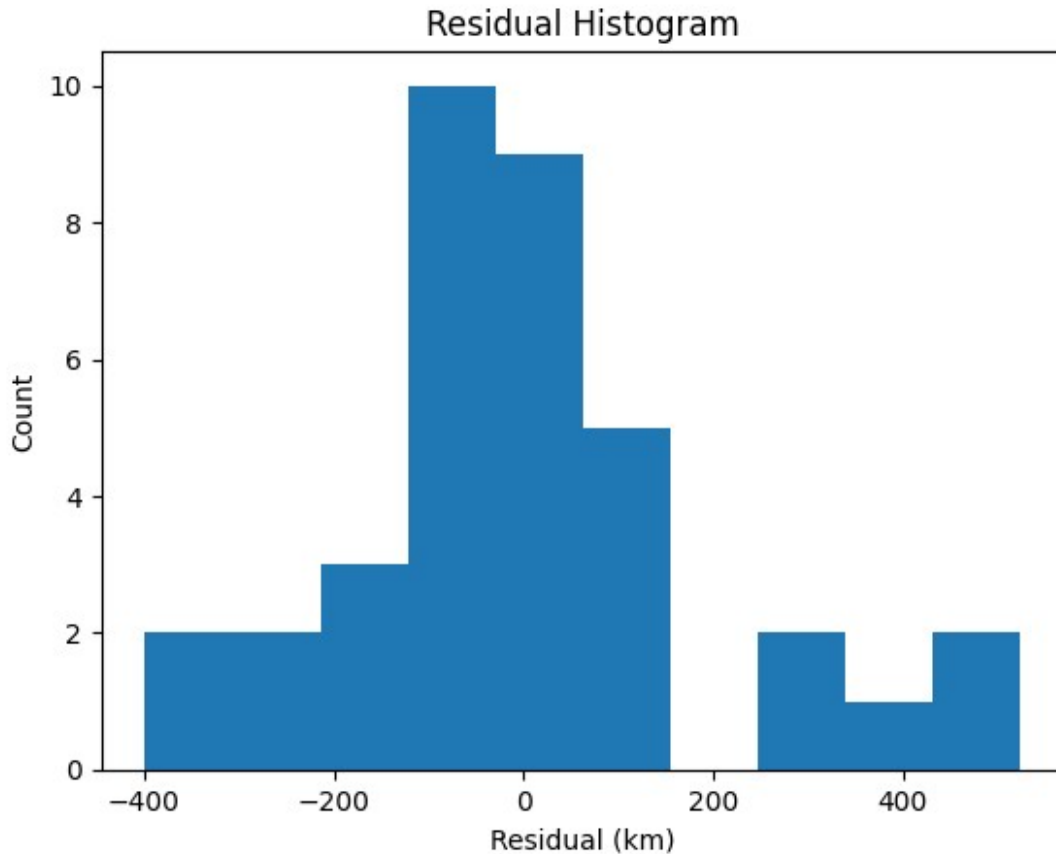
```
plt.xlabel("Age (Myr)")
plt.ylabel("Residual (km)")
plt.title("Residuals vs Age")
plt.show()
```

```
# Histogram
plt.figure()
plt.hist(residuals, bins=10)
plt.xlabel("Residual (km)")
plt.ylabel("Count")
plt.title("Residual Histogram")
plt.show()
```

Goodness of Fit

RMSE: 200.9056 km
 R^2 : 0.983539





Gaussian Uncertainty & 95% Confidence Interval

Under the Gaussian noise assumption:

$$\epsilon_i \sim N(0, \sigma^2)$$

Least Squares is Maximum Likelihood Estimation.

We estimate:

$$\sigma^2_{\text{hat}} = (1 / (N - 2)) * \text{sum}(\text{residuals}^2)$$

Then the parameter covariance:

$$\text{Cov}(\mathbf{w}_{\text{hat}}) = \sigma^2_{\text{hat}} * (\mathbf{X}^T \mathbf{X})^{(-1)}$$

From this we compute:

- Standard error of velocity
- 95% confidence interval

This quantifies how certain we are about Pacific Plate velocity.

```

# Degrees of freedom (2 parameters: intercept + slope)
dof = N - 2

# Estimate noise variance
sigma2_hat = np.sum(residuals**2) / dof

# Parameter covariance matrix
cov_w = sigma2_hat * XtX_inv

# Standard errors
se_b = np.sqrt(cov_w[0, 0])
se_v = np.sqrt(cov_w[1, 1])

# 95% CI
z = 1.96

v_lower = v_hat - z * se_v
v_upper = v_hat + z * se_v

print("Uncertainty Analysis")
print()
print(f"Estimated noise variance  $\sigma^2$ : {sigma2_hat:.4f}")
print(f"Std Error (velocity): {se_v:.4f} km/Myr")
print(f"95% CI for velocity: [{v_lower:.4f}, {v_upper:.4f}] km/Myr")

# Convert to cm/year for intuition
v_cm_per_year = v_hat * 0.1
v_lower_cmy = v_lower * 0.1
v_upper_cmy = v_upper * 0.1

print("\nVelocity in cm/year:")
print(f"{v_cm_per_year:.3f} cm/year")
print(f"95% CI: [{v_lower_cmy:.3f}, {v_upper_cmy:.3f}] cm/year")

Uncertainty Analysis

Estimated noise variance  $\sigma^2$ : 42737.3587
Std Error (velocity): 1.6833 km/Myr
95% CI for velocity: [72.5716, 79.1703] km/Myr

Velocity in cm/year:
7.587 cm/year
95% CI: [7.257, 7.917] cm/year

```

Final Interpretation: Model Validity, Physical Meaning & Limitations

Estimated Pacific Plate Velocity

From our Least Squares + Gaussian model:

$$\begin{aligned}\hat{v} &= 75.871 \text{ km/Myr} \\ &= 7.587 \text{ cm/year}\end{aligned}$$

95% Confidence Interval:

$$\begin{aligned}[72.57, 79.17] &\text{ km/Myr} \\ = [7.257, 7.917] &\text{ cm/year}\end{aligned}$$

This interval is tight relative to the magnitude of the estimate, indicating strong statistical confidence in the velocity.

Model Quality

- $R^2 \approx 0.9835 \rightarrow \sim 98.35\%$ of variance explained.
- Residuals centered near zero.
- No strong nonlinear pattern in residual plot.
- Histogram approximately bell-shaped.

These support:

- Linear drift assumption.
- Gaussian noise assumption.
- Least Squares \approx Maximum Likelihood validity.

Physical Interpretation

The Hawaiian-Emperor chain provides geological evidence of Pacific Plate motion over a relatively stationary hotspot.

Our estimate (~ 7.6 cm/year) aligns well with independent geophysical plate motion estimates, which typically range between 7–10 cm/year.

The nonzero intercept (~ 221 km) likely reflects:

- Imperfect hotspot reference point.
- Geological uncertainty.
- Model simplification (ignoring hotspot migration or plate direction change).

Limitations

1. Assumes constant velocity over ~ 65 Myr.
2. Assumes stationary hotspot.
3. Assumes independent Gaussian noise.
4. Does not model directional changes (e.g., Emperor bend).

Possible Extensions

- Weighted Least Squares (if age uncertainty varies).
- Piecewise linear model (detect change in plate direction).
- Bayesian inference on velocity.
- Robust regression (Huber loss).
- Full probabilistic model with uncertainty propagation.

Final Conclusion

A simple linear Gaussian drift model provides a statistically strong and physically plausible estimate of Pacific Plate velocity.

This demonstrates:

- Proper use of Least Squares.
- Understanding of probabilistic modeling.
- Model validation via residual diagnostics.
- Quantified uncertainty.

The model is both mathematically sound and geophysically meaningful.
