```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Given data
time_taken = np.array([60, 27, 20, 76, 30, 25])
hf = np.array([2.81*12.6, 4.85*12.6, 6.33*12.6, 2.57*12.6, 4.21*12.6, 5.46*12.6])
r = 20
rho_water = 1000
g = 981
L = 150
A = 1172
D = 2.15
a = 3.3605

# Calculate Qactual and Qtheoretical
Qactual = A * r / time_taken
V = Qactual / time_taken
f1 = (hf * 2 * g * D) / (4 * L * V ** 2)

# Calculate logarithms
X = np.log10(V)
Y = np.log10(hf)
A = X ** 2
B = Y ** 2
C = X * Y

# Calculate sums
Sxy = np.sum(X * Y) - np.sum(X) * np.sum(Y) / 6
Sxx = np.sum(X ** 2) - (np.sum(X) ** 2) / 6
Syy = np.sum(Y ** 2) - (np.sum(Y) ** 2) / 6

# Calculate n, k, and K
n = Sxy / Sxx
k = 10 ** ((np.sum(Y) - n * np.sum(X)) / 6)
K = (np.sum(Y) - n * np.sum(X)) / 6
mean_f1 = np.mean(f1)

# Create a DataFrame for all variables
results_data = {
    'Qactual': Qactual,
    'Velocity': V,
    'friction': f1,
    'X': X,
    'Y': Y,
    'X*X': A,
    'Y*Y': B,
    'X*Y': C,
}
results_df = pd.DataFrame(results_data)
```

```
# Display the DataFrame
print(results_df)

# Print remaining values
print('f1:', mean_f1)
print('Sxy:', Sxy)
print('Sxx:', Sxx)
print('Syy:', Syy)
print('n:', n)
print('k:', k)
print('K:', K)

# Find the slope of the line passing through the maximum number of points
coefficients = np.polyfit(X, Y, 1)
slope = coefficients[0]

# Calculate x-intercept (X) and y-intercept (Y)
x_intercept = coefficients[1] / coefficients[0]
y_intercept = coefficients[1]

# Create the plot
plt.figure()
plt.plot(X, Y, 'o', X, slope * X + y_intercept, 'r', linewidth=2)
plt.xlabel('log10(V)')
plt.ylabel('log10(hf)')
plt.title('Log-Log Plot of hf vs. V')
plt.grid(True)

# Display the slope, x-intercept, and y-intercept
print('Slope (m):', slope)
print('X-intercept (X):', x_intercept)
print('Y-intercept (Y):', y_intercept)

# Calculate Cd2
f2 = (K * 2 * g * D)/(4 * L)
print('f2:', f2)

plt.show()
```