```python
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import scipy.stats as stats
from sklearn.linear_model import LinearRegression

def main():
    # Import the data
    walking = pd.read_csv("walking.csv")
    print(walking)

    # Make a cut of the first walk away
    walking = walking.head(146)
    print(walking)

    # Convert relevant pandas crap to numpy arrays
    walk_time = walking["Time (s) Run #1"].values.reshape(-1, 1)
    walk_pos = walking["Position (m) Run #1"].values.reshape(-1, 1)
    linreg = LinearRegression()
    lin_model = linreg.fit(walk_time, walk_pos)
    fit = linreg.predict(walk_time)
    print(f"f(x)={linreg.coef_}x+{linreg.intercept_}")

    # Create a plot of the data
    plt.scatter(
        walk_time,
        walk_pos,
        color="dodgerblue",
        label="collected data"
    )
    plt.plot(
        walk_time,
        fit,
        color='red',
        label="Regression Line"
    )
    plt.xlabel("Time (sec)")
    plt.ylabel("Position (meters)")
    plt.title("Position vs Time of a Student Walking")
    plt.show()

    # import the running data
    running = pd.read_csv("running.csv")
    print(running)

    # Make a quality cut
    running = running.head(46)
    print(running)

    # do a regression
    run_time = running["Time (s) Run #1"]
    run_pos = running["Position (m) Run #1"]

    quad_model = np.poly1d(np.polyfit(run_time, run_pos, 2))
    print(quad_model)

    # Create a plot of the data
    plt.scatter(
        run_time,
        run_pos,
        color='dodgerblue',
        label="Collected Data"
    )
    plt.plot(
        run_time,
        quad_model(run_time),
        color='red',
        label="Quadratic Model"
    )
    plt.title("Kinematics of a Running Student")
    plt.xlabel("Time (sec)")
    plt.ylabel("Position (meters)")
    plt.legend()
    plt.show()

    # Find the average acceleration using the collected data
    avg_accel = running.mean(axis=0)
    print(f"Mean Acceleration: {avg_accel}")

main()
```