```
## Acceleration Mean Stats for FACE UP
### I DO NOT CHANGE VARIABLES NAMES BUT RATHER PLACE A NOTE AT THE BEGINNING INDICATING WHA
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalu
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACEUP = [
[0.29, 0.01, 0.95],
[0.31, 0.00, 0.95],
[0.31, 0.00, 0.95],
[0.30, 0.02, 0.94],
[0.29, 0.03, 0.97],
[0.27, 0.06, 0.95],
[0.06, -0.06, 0.90],
[0.05, 0.08, 0.98],
[0.06, 0.03, 0.99],
[0.08, 0.02, 0.99],
[0.06, 0.07, 1.01],
[0.08, 0.06, 0.98],
[0.09, 0.09, 0.98],
[0.09, 0.05, 0.99],
[0.08, 0.05, 0.99],
[0.08, 0.05, 1.00],
[0.07, 0.03, 1.03],
[0.09, 0.06, 1.02],
[0.09, 0.05, 1.01],
[0.10, 0.05, 0.99],
[0.10, -0.01, 1.00],
[0.10, 0.02, 1.02],
[0.12, 0.01, 0.97],
[0.10, 0.05, 0.96],
[0.09, 0.05, 1.01],
[0.11, 0.03, 0.97],
[0.10, 0.01, 1.00],
[0.11, 0.02, 1.01],
[0.12, 0.05, 0.98],
[0.11, 0.06, 0.98],
[0.12, 0.07, 0.96],
[0.12, 0.06, 0.98],
[0.11, 0.02, 1.00],
[0.11, 0.03, 1.00],
[0.11, 0.04, 0.97],
[0.11, 0.07, 0.97],
[0.12, 0.06, 0.99],
[0.12, 0.04, 1.00],
[0.12, 0.01, 0.99],
[0.12, 0.07, 0.96],
[0.14, 0.03, 0.93]
```

```
]



# Convert your data to a NumPy array for easy calculations.
data_array1 = np.array(data_accel_FACEUP)

# Calculate the mean and standard deviation for each axis.
x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

# Print the calculated means and standard deviations.
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 0.1270731707317073
Mean for 'y' axis: 0.0375609756097561
Mean for 'z' axis: 0.9809756097560975
Standard deviation for 'x' axis: 0.0724230088232989
Standard deviation for 'y' axis: 0.02817891339324133
Standard deviation for 'z' axis: 0.026207881708611786
```

```
## Acceleration Mean Stats for FACE SIDE
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalu
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACESIDE = [
[0.26, -0.96, 0.12],
[0.28, -0.93, 0.06],
[0.27, -0.97, 0.08],
[0.27, -0.95, 0.08],
[0.28, -0.96, 0.06],
[0.29, -0.96, 0.05],
[0.28, -0.99, 0.08],
[0.30, -0.96, 0.04],
[0.29, -0.95, 0.05],
[0.25, -0.96, 0.04],
[0.24, -1.00, -0.02],
[0.27, -1.06, -0.07],
[0.22, -0.95, 0.01]
```

```
[0.22, -0.95, 0.01],
[0.25, -0.93, 0.04],
[0.25, -0.99, -0.06],
[0.26, -0.99, -0.02],
[0.24, -1.00, -0.06],
[0.26, -1.00, -0.02],
[0.28, -0.97, -0.04],
[0.26, -0.97, -0.10],
[0.23, -0.98, -0.04],
[0.24, -0.98, -0.04],
[0.24, -0.99, -0.06],
[0.25, -0.97, -0.04],
[0.25, -1.00, -0.07],
[0.26, -0.98, -0.05],
[0.26, -0.96, -0.07],
[0.27, -0.97, -0.08],
[0.27, -0.97, -0.09],
[0.23, -0.97, -0.07],
[0.26, -0.97, -0.07],
[0.26, -0.97, -0.10],
[0.27, -1.01, -0.08]

]

# Convert your data to a NumPy array for easy calculations.
data_array1 = np.array(data_accel_FACESIDE)

# Calculate the mean and standard deviation for each axis.
x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

# Print the calculated means and standard deviations.
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 0.2603030303030303
Mean for 'y' axis: -0.9748484848484846
Mean for 'z' axis: -0.016363636363636365
Standard deviation for 'x' axis: 0.018337715479281542
Standard deviation for 'y' axis: 0.024261356243937094
Standard deviation for 'z' axis: 0.062319712422907
```

```
## Acceleration Mean Stats for FACE DOWN
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalı
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACEDOWN = [
[-0.33, -0.21, -0.93],
[-0.33, -0.16, -0.96],
[-0.29, -0.05, -0.92],
[-0.19, -0.01, -1.03],
[-0.04, -0.02, -0.97],
[0.05, -0.02, -1.03],
[0.08, 0, -1.05],
[0.05, -0.04, -1],
[0.09, -0.06, -1.03],
[0.09, -0.08, -1.03],
[0.09, -0.06, -1.02],
[0.09, -0.07, -0.96],
[0.13, -0.11, -1],
[0.14, -0.08, -1.03],
[0.15, -0.07, -0.96],
[0.17, -0.12, -1.02],
[0.16, -0.05, -0.99],
[0.14, -0.07, -0.99],
[0.13, -0.09, -1.05],
[0.13, -0.03, -1.02],
[0.12, -0.01, -1.01],
[0.12, 0, -1.02],
[0.11, 0, -1],
[0.12, 0.01, -1.03],
[0.11, 0.01, -1.02],
[0.12, 0, -1.01],
[0.10, 0.02, -1],
[0.11, 0.02, -1.01],
[0.10, -0.01, -1],
[0.11, -0.01, -1.03],
[0.12, 0, -1.05],
[0.10, 0, -1.02],
[0.10, 0.02, -1.01],
[0.10, 0.01, -1.03],
[0.10, 0.01, -1.01],
[0.08, 0, -1.01],
[0.09, -0.01, -1.01],
[0.09, 0, -1.03],
[0.09, -0.03, -1.02],
[0.09, -0.04, -1],
[0.08, 0.01, -0.98]

]
```

```
# Convert your data to a NumPy array for easy calculations.
data_array1 = np.array(data_accel_FACEDOWN)

# Calculate the mean and standard deviation for each axis.
x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

# Print the calculated means and standard deviations.
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 0.0651219512195122
Mean for 'y' axis: -0.03414634146341464
Mean for 'z' axis: -1.0070731707317073
Standard deviation for 'x' axis: 0.12117558216598717
Standard deviation for 'y' axis: 0.05041174900590756
Standard deviation for 'z' axis: 0.029070469673186085
```

```
## Acceleration Mean Stats for GYRO FACE UP
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalu
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACEDOWN = [
[2.38, 0, 0.55],
[1.77, 1.59, 0.67],
[0.49, 0.31, -0.43],
[1.65, 1.4, 1.65],
[1.4, -0.92, 0.61],
[-5.62, -5.86, 0.98],
[0.49, -1.53, -0.31],
[-2.62, -0.61, 7.45],
[1.16, 0.92, -5.49],
[1.46, -1.22, -1.46],
[1.4, 0.06, 3.66],
[0.55, 3.17, 2.2],
[1.71, 1.16, 1.83],
[1.34, 1.16, -2.2],
[2.14, 3.23, -3.54],
[-1.04, 1.89, -0.61],
```

```
[1.83, -0.67, 4.09],
[2.14, 0.37, 4.33],
[3.17, -0.98, -0.55],
[1.71, 2.69, -2.26],
[1.46, 1.28, -0.73],
[1.53, -0.12, 2.26],
[4.03, -1.16, 0.12],
[-0.73, 1.1, 1.22],
[2.56, -0.43, 0.24],
[2.62, 0.18, -0.37],
[0.79, 1.65, -1.83],
[1.22, 2.32, 0.98],
[2.14, 1.1, 2.32],
[0.55, -0.37, 1.4],
[2.14, 2.44, 0.37],
[2.26, -0.79, -1.16],
[1.34, 0.37, 0.31],
[1.95, 0.98, -1.53],
[1.22, -0.31, 1.77],
[1.04, 1.34, 2.81],
[2.99, -0.31, -1.4],
[2.87, 1.83, -0.98],
[0.79, 1.95, 1.46],
[0.79, 0.18, 0.24],
[5.43, -0.37, 1.59]


]

# Convert your data to a NumPy array for easy calculations.
data_array1 = np.array(data_accel_FACEDOWN)

# Calculate the mean and standard deviation for each axis.
x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

# Print the calculated means and standard deviations.
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 1.378048780487805
```

```
Mean for 'x' axis: 1.3780487804878O5
Mean for 'y' axis: 0.4639024390243902
Mean for 'z' axis: 0.4941463414634146
Standard deviation for 'x' axis: 1.7036219944751503
Standard deviation for 'y' axis: 1.5802637600911644
Standard deviation for 'z' axis: 2.2341102617174036
```

```
## Acceleration Mean Stats for GYRO FACE SIDE
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalu
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACEDOWN = [
[1.53, 0, 1.16],
[3.54, -0.67, 1.28],
[2.01, -0.12, 1.4],
[3.42, 1.1, 1.83],
[2.62, 0, -0.18],
[2.32, 0.12, 0],
[3.11, 0.49, -0.31],
[2.87, 1.1, 1.1],
[3.3, 1.83, -1.4],
[10.56, -0.92, -2.08],
[0.24, -3.36, 1.65],
[2.44, 2.14, 2.62],
[-2.32, 2.14, 2.56],
[4.76, 3.05, 5],
[1.34, 0.79, 0.49],
[1.4, -1.1, -0.55],
[1.77, 1.16, 2.2],
[2.08, -2.14, 1.53],
[0.92, -0.67, -1.1],
[3.36, 5.07, -1.16],
[1.71, 0.73, 0.79],
[1.77, -0.12, -0.31],
[1.1, 1.71, 0.06],
[2.26, 0.12, -0.18],
[2.32, 0.37, 0.49],
[4.39, 0.12, 0],
[2.87, 1.04, 0.06],
[2.93, 1.22, -1.46],
[2.81, 1.53, -0.12],
[0.85, 0, 0.43],
[2.32, -0.43, -0.49],
[3.66, 0.43, -0.12],
[3.05, -2.44, -0.79],
[3.05, -1.28, 0.31],
[2.5, 1.77, -0.06],
[0.79, -0.12, -0.49],
[0.92, -0.06, 0.24],
[2.69, 0.24, 2.26],
```

```
[4.52, 1.95, -0.12],
[1.22, -0.06, 1.22],
[-2.26, -1.4, 3.85]



]

# Convert your data to a NumPy array for easy calculations.
data_array1 = np.array(data_accel_FACEDOWN)

# Calculate the mean and standard deviation for each axis.
x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

# Print the calculated means and standard deviations.
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 2.359512195121951
Mean for 'y' axis: 0.37390243902439035
Mean for 'z' axis: 0.5270731707317073
Standard deviation for 'x' axis: 1.9389888629971264
Standard deviation for 'y' axis: 1.4877546972745015
Standard deviation for 'z' axis: 1.4141018763148798
```

```
## Acceleration Mean Stats for GYRO FACE DOWN
import numpy as np
## For each posture, we collected 50 data points for accelerometer & gyroscopic data. evalu
# Replace this data with your accelerometer data for each axis.
# Make sure 'data' is a list or NumPy array containing your data.
data_accel_FACEDOWN = [
[1.22, -0.06, 0.49],
[5.92, 2.14, 1.4],
[8.79, -4.21, 2.81],
[8.67, -9.09, 2.2],
[-1.16, -4.15, -0.18],
[7.75, -5.07, -8.73],
[4.82, -4.88, -1.1],
[-0.55, 1.04, 2.56],
```

```
    [2.2, -0.98, 2.26],
    [2.75, -0.92, 1.04],
    [0.18, -0.12, -1.04],
    [1.83, -2.87, 3.05],
    [-0.24, -1.77, 1.71],
    [0.79, 1.4, -1.71],
    [1.89, -0.49, 1.83],
    [3.72, -0.98, 1.04],
    [1.4, 1.4, -1.59],
    [4.76, -0.85, 0.79],
    [3.66, -1.04, 0],
    [5.98, 1.77, 0.85],
    [3.05, 0.55, -0.67],
    [1.71, -1.65, 0.37],
    [2.14, 0, -0.55],
    [3.42, 1.59, 1.28],
    [1.28, 0.06, -0.67],
    [3.17, -2.87, -0.24],
    [1.16, -0.18, 1.46],
    [1.95, -0.24, -0.37],
    [3.05, -0.12, 0.24],
    [0.73, -0.24, 0.37],
    [1.95, 2.32, -0.18],
    [1.89, 1.89, 0.31],
    [0.98, 0.92, -1.28],
    [1.59, 1.04, 0.12],
    [2.5, 1.71, -0.37],
    [1.22, 0.43, -0.73],
    [1.4, 0.12, -0.12],
    [2.44, -0.31, 0.24],
    [1.53, -2.01, 0.55],
    [2.32, -1.16, 2.5],
    [2.5, -0.67, 0.24]




    ]

    # Convert your data to a NumPy array for easy calculations.
    data_array1 = np.array(data_accel_FACEDOWN)

    # Calculate the mean and standard deviation for each axis.
    x_mean = np.mean(data_array1[:, 0])  # Mean for the 'x' axis (column 0).
    y_mean = np.mean(data_array1[:, 1])  # Mean for the 'y' axis (column 1).
    z_mean = np.mean(data_array1[:, 2])  # Mean for the 'z' axis (column 2).

    x_stddev = np.std(data_array1[:, 0])  # Standard deviation for the 'x' axis (column 0).
    y_stddev = np.std(data_array1[:, 1])  # Standard deviation for the 'y' axis (column 1).
    z_stddev = np.std(data_array1[:, 2])  # Standard deviation for the 'z' axis (column 2).

    # Print the calculated means and standard deviations.
```

```
# Print the calculated means and standard deviations:
print("Mean for 'x' axis:", x_mean)
print("Mean for 'y' axis:", y_mean)
print("Mean for 'z' axis:", z_mean)
print("Standard deviation for 'x' axis:", x_stddev)
print("Standard deviation for 'y' axis:", y_stddev)
print("Standard deviation for 'z' axis:", z_stddev)
```

```
Mean for 'x' axis: 2.5941463414634147
Mean for 'y' axis: -0.6963414634146342
Mean for 'z' axis: 0.2482926829268292
Standard deviation for 'x' axis: 2.2158453494615813
Standard deviation for 'y' axis: 2.2632749378673256
Standard deviation for 'z' axis: 1.8595408591960627
```