In [1]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import Perceptron
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix,classification_repor
t
import matplotlib.pyplot as plt
```

In [56]:

```
df = pd.read_csv('../FeaturesCsvFile/featuresfile.csv')
df.shape
```

Out[56]:

```
(417, 46)
```

In [16]:

```
X = df.values[:, 2:45]
y = df.values[:, 45] #label : walking/runing
y_plot = np.where(y == 'walking', -1, 1)
X_train, X_test, y_train, y_test = train_test_split(X, y_plot, test_size=0.3)
ppn = Perceptron(max_iter=40, eta0=0.1, random_state=1)
ppn.fit(X_train, y_train)
y_pred = ppn.predict(X_test)
```

In [17]:

```
print('Accuracy of Accuracy Score : %.2f' % accuracy_score(y_test,y_pred))
```

```
Accuracy of Accuracy Score : 0.98
```

In [18]:

```
print('Accuracy of Perceptron Score: %.2f' % ppn.score(X_test,y_test))
```

```
Accuracy of Perceptron Score: 0.98
```

In [19]:

```
print ('Confusion_matrix')
print(confusion_matrix(y_test,y_pred))
```

```
Confusion_matrix
[[71  1]
 [ 1 53]]
```

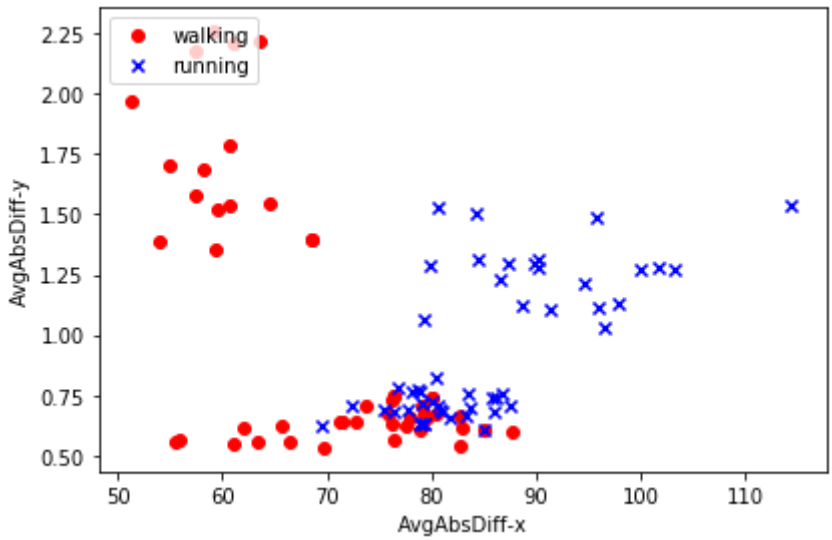
In [20]:

```
print ('Important features')
header = list(df.head(1))
for i in range(0,len(ppn.coef_[0])):
    print (i+1,header[i+2],ppn.coef_[0][i])
```

```
Important features
(1, 'Bin1,x', 0.24709872018647222)
(2, 'Bin2,x', 0.90639319829669329)
(3, 'Bin3,x', 1.569215071314277)
(4, 'Bin4,x', 0.87659842412923306)
(5, 'Bin5,x', -0.71113074885140781)
(6, 'Bin6,x', -1.6620896870166044)
(7, 'Bin7,x', 0.18472626165352474)
(8, 'Bin8,x', 3.1187938892078164)
(9, 'Bin9,x', 3.3452381497143056)
(10, 'Bin10,x', 1.3888095686454358)
(11, 'Bin1,y', 0.57132153462569002)
(12, 'Bin2,y', 2.2066329026069278)
(13, 'Bin3,y', 2.5745691003965629)
(14, 'Bin4,y', 1.7854557978072789)
(15, 'Bin5,y', 1.9186120294028235)
(16, 'Bin6,y', 2.0984350748453888)
(17, 'Bin7,y', 1.048837686881108)
(18, 'Bin8,y', -0.53506441244885794)
(19, 'Bin9,y', -0.91675046488809231)
(20, 'Bin10,y', -0.721703767208692)
(21, 'Bin1,z', 0.1549625422581109)
(22, 'Bin2,z', -0.33051928805275771)
(23, 'Bin3,z', 0.22017900856118305)
(24, 'Bin4,z', 2.306293060803692)
(25, 'Bin5,z', 3.1043382088926923)
(26, 'Bin6,z', 2.4935516646305436)
(27, 'Bin7,z', 1.3698331156919306)
(28, 'Bin8,z', -0.037041144265181403)
(29, 'Bin9,z', -0.096127912347931721)
(30, 'Bin10,z', 0.30932026826460979)
(31, 'TimeDiffPeaks-x', 31.581449672500003)
(32, 'TimeDiffPeaks-y', -24.085878239889958)
(33, 'TimeDiffPeaks-z', -37.172866016200025)
(34, 'AvgAbsDiff-x', 112.93846207906782)
(35, 'AvgAbsDiff-y', 82.969349889883858)
(36, 'AvgAbsDiff-z', 94.877565338729823)
(37, 'AvgAcc-x', -20.489632786484197)
(38, 'AvgAcc-y', -50.510837138510865)
(39, 'AvgAcc-z', 118.01404728085981)
(40, 'StdDev-x', 132.37990056954391)
(41, 'StdDev-y', 100.41326445323827)
(42, 'StdDev-z', 118.21791536124083)
(43, 'AvgResAcc', 7.2770710888601462)
```

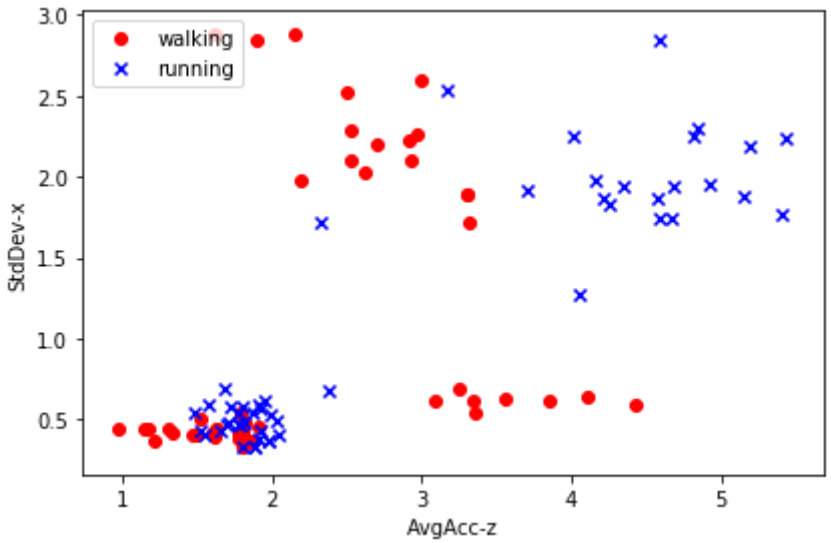
In [22]:

```
#get data first 100 rows
y = df.iloc[0:100, 45].values
y = np.where(y == 'walking', -1, 1)
#use AvgAbsDiff-x and AvgAbsDiff-y
X = df.iloc[0:100, [34, 36]].values
# plot data
plt.scatter(X[:50, 0], X[:50, 1],
            color='red', marker='o', label='walking')
plt.scatter(X[50:100, 0], X[50:100, 1],
            color='blue', marker='x', label='running')

plt.xlabel('AvgAbsDiff-x')
plt.ylabel('AvgAbsDiff-y')
plt.legend(loc='upper left')
plt.tight_layout()
plt.show()
```



In [42]:

```
#get data first 100 rows
y = df.iloc[0:100, 45].values
y = np.where(y == 'walking', -1, 1)
#use AvgAcc-z and StdDev-x
X = df.iloc[0:100, [39, 41]].values
# plot data
plt.scatter(X[:50, 0], X[:50, 1],
            color='red', marker='o', label='walking')
plt.scatter(X[50:100, 0], X[50:100, 1],
            color='blue', marker='x', label='running')

plt.xlabel('AvgAcc-z')
plt.ylabel('StdDev-x')
plt.legend(loc='upper left')
plt.tight_layout()
plt.show()
```



In [62]:

```
y = df.iloc[:len(df), 45].values
y = np.where(y == 'walking', -1, 1)
#use
X = df.iloc[:len(df), [2,45]].values
# plot data
plt.scatter(X[:len(df)/2, 0], X[:len(df)/2, 1],
            color='red', marker='o', label='walking')
plt.scatter(X[len(df)/2:len(df), 0], X[len(df)/2:len(df), 1],
            color='blue', marker='x', label='running')

plt.legend(loc='upper left')
plt.tight_layout()
plt.show()
```