kstest_walk_run

December 11, 2017

```
In [1]: %matplotlib inline
        from pathlib import Path
        from sklearn import tree
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.metrics import accuracy_score
        from sklearn.metrics import confusion_matrix
        from scipy import stats
        import numpy as np
        import json
```

```
In [2]: filePath_walk = Path("./../DecisionTreeClassifier/Data/surada_walking_10.json")
        filePath_run = Path("./../DecisionTreeClassifier/Data/surada_running_10.json")
        f = open(filePath_walk)
        f_run = open(filePath_run)
        line = f.readline()
        line_run = f_run.readline()
        running_json = json.loads(line_run)
        walking_json = json.loads(line)
        x_walk = walking_json['xAxis']
        y_walk = walking_json['yAxis']
        z_walk = walking_json['zAxis']
        x_run = running_json['xAxis']
        y_run = running_json['yAxis']
        z_run = running_json['zAxis']
```

```
In [3]: x_kswalk = stats.kstest(x_walk, 'norm')
        y_kswalk = stats.kstest(y_walk, 'norm')
        z_kswalk = stats.kstest(z_walk, 'norm')
```

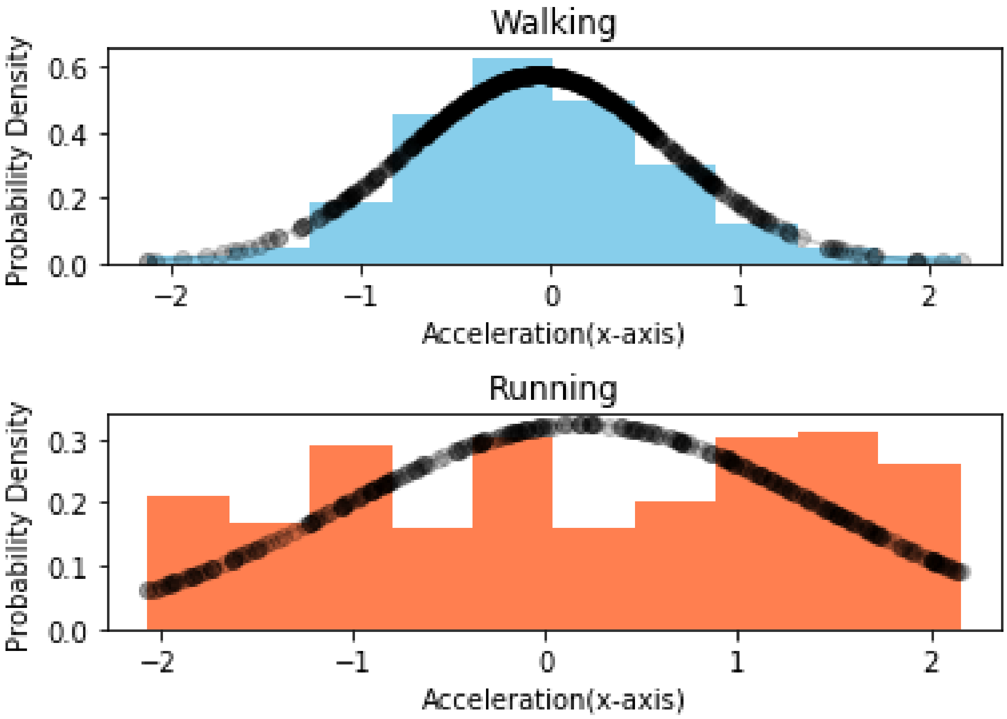
```
In [4]: x_kswalk
```

```
Out[4]: KstestResult(statistic=0.12616951523215136, pvalue=1.5052580293328788e-08)
```

```
In [5]: y_kswalk
```

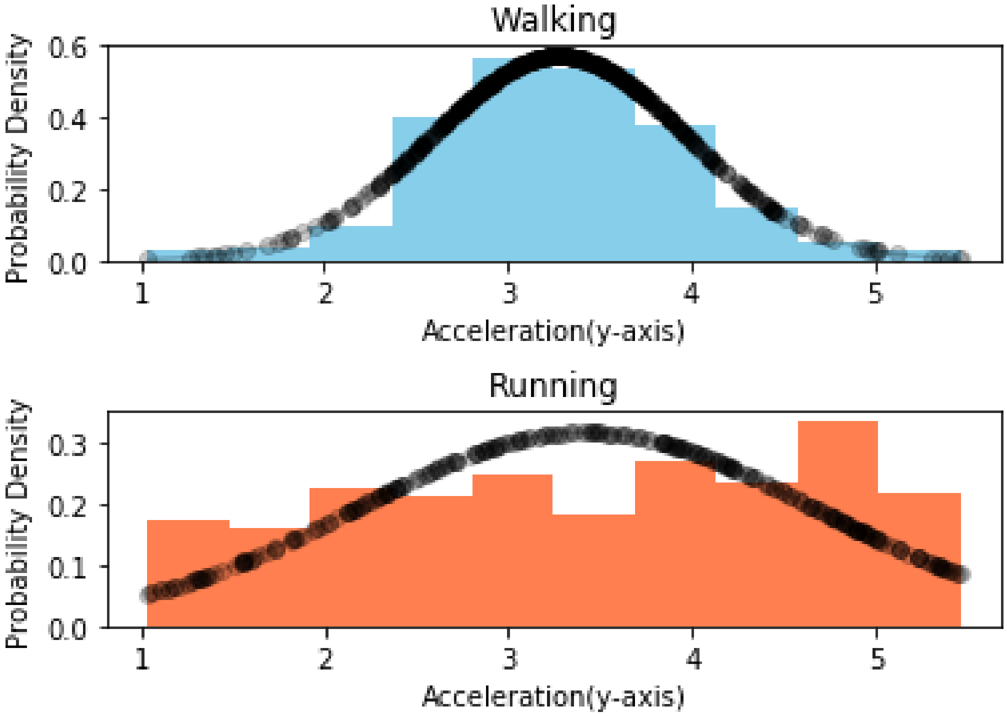
```
Out[5]: KstestResult(statistic=0.94862298011537638, pvalue=0.0)
```

```
In [6]: z_kswalk

Out[6]: KstestResult(statistic=0.99999996155595661, pvalue=0.0)

In [7]: x_ksrun = stats.kstest(x_run, 'norm')
        y_ksrun = stats.kstest(y_run, 'norm')
        z_ksrun = stats.kstest(z_run, 'norm')

In [8]: x_ksrun

Out[8]: KstestResult(statistic=0.24717740770457636, pvalue=0.0)

In [9]: y_ksrun

Out[9]: KstestResult(statistic=0.65795225220628384, pvalue=0.0)

In [10]: z_ksrun

Out[10]: KstestResult(statistic=0.97796180299629354, pvalue=0.0)

In [11]: from scipy.stats import ks_2samp

         ks_2samp(x_walk,x_run)

Out[11]: Ks_2sampResult(statistic=0.31657055349730356, pvalue=7.2452503618006156e-26)

In [12]: ks_2samp(x_walk,x_walk)#for identical distributions statistic would be 0 and pvalue w

Out[12]: Ks_2sampResult(statistic=0.0, pvalue=1.0)

In [13]: ks_2samp(y_walk,y_run)

Out[13]: Ks_2sampResult(statistic=0.32083008015627301, pvalue=1.4819023215773932e-26)

In [14]: ks_2samp(z_walk,z_run)

Out[14]: Ks_2sampResult(statistic=0.29413041446731802, pvalue=2.1826951305691418e-22)

In [27]: import numpy as np
         import scipy.stats as stats
         import pylab as pl
         import matplotlib.pyplot as plt

         pl.subplot(2,1,1)
         sort_x_walk = sorted(x_walk)  #sorted
         fit = stats.norm.pdf(sort_x_walk, np.mean(sort_x_walk), np.std(sort_x_walk))  #this i
         pl.plot(sort_x_walk,fit,'-o',color='black',alpha=0.2)
         pl.xlabel('Acceleration(x-axis)')
         pl.ylabel('Probability Density')
         pl.hist(sort_x_walk,normed=True,color='skyblue')#use this to draw histogram of your d
         pl.title('Walking')
         pl.subplot(2,1,2)
         sort_x_run = sorted(x_run)   #sorted
         sort_x_run_np = np.array(sort_x_run)
         indices_arr = np.where(np.logical_and(sort_x_run_np>=sort_x_walk[0], sort_x_run_np<=so
         first = indices_arr[0][0]
         last = indices_arr[0][len(indices_arr[0])-1]
         sort_x_run_mod = sort_x_run_np[first:last]
         fit = stats.norm.pdf(sort_x_run_mod, np.mean(sort_x_run_mod), np.std(sort_x_run_mod))
         pl.plot(sort_x_run_mod,fit,'-o',color='black',alpha=0.2)
         pl.hist(sort_x_run_mod,normed=True,color='coral')       #use this to draw histogram of
         pl.xlabel('Acceleration(x-axis)')
         pl.ylabel('Probability Density')
         pl.title('Running')
         pl.subplots_adjust(hspace=0.7)
         pl.savefig("./../Data-Visualization/images/X-distributions.png",dpi=1000)
```



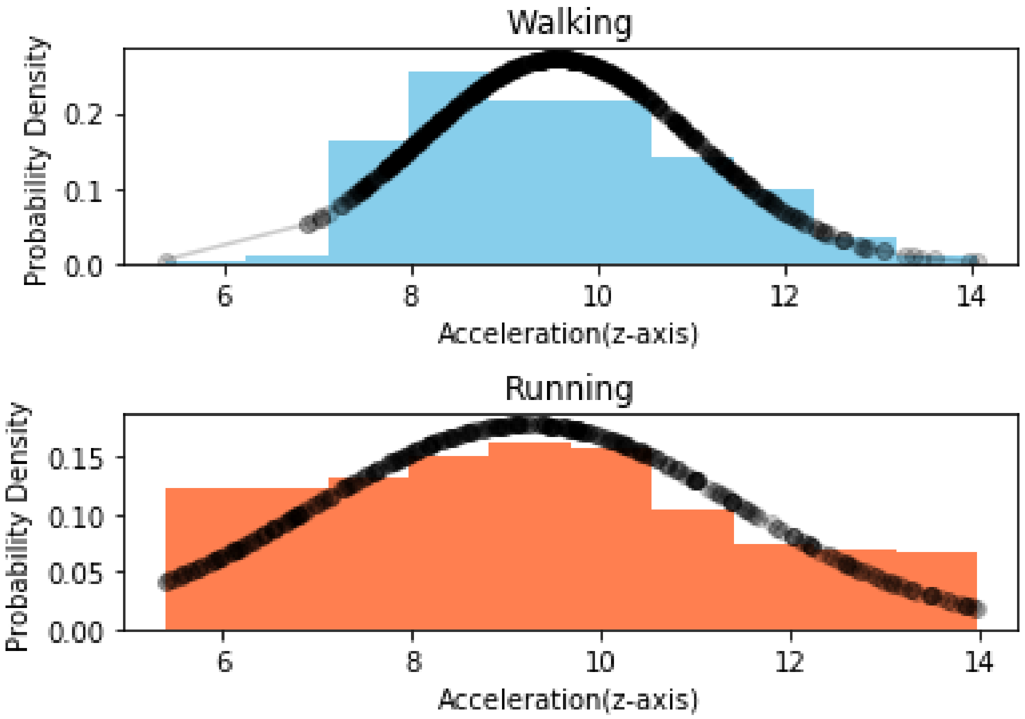
```
In [28]: import numpy as np
         import scipy.stats as stats
         import pylab as pl
         import matplotlib.pyplot as plt

         pl.subplot(2,1,1)
         sort_y_walk = sorted(y_walk)   #sorted
```

```
fit = stats.norm.pdf(sort_y_walk, np.mean(sort_y_walk), np.std(sort_y_walk))  #this i
pl.plot(sort_y_walk,fit,'-o',color='black',alpha=0.2)
pl.xlabel('Acceleration(y-axis)')
pl.ylabel('Probability Density')
pl.title('Walking')
pl.hist(sort_y_walk,normed=True,color='skyblue')#use this to draw histogram of your d

pl.subplot(2,1,2)
sort_y_run = sorted(y_run)  #sorted
sort_y_run_np = np.array(sort_y_run)
indices_arr = np.where(np.logical_and(sort_y_run_np>=sort_y_walk[0], sort_y_run_np<=so
first = indices_arr[0][0]
last = indices_arr[0][len(indices_arr[0])-1]
sort_y_run_mod = sort_y_run_np[first:last]
fit = stats.norm.pdf(sort_y_run_mod, np.mean(sort_y_run_mod), np.std(sort_y_run_mod))
pl.plot(sort_y_run_mod,fit,'-o',color='black',alpha=0.2)
pl.hist(sort_y_run_mod,normed=True,color='coral')      #use this to draw histogram of
pl.xlabel('Acceleration(y-axis)')
pl.ylabel('Probability Density')
pl.title('Running')
pl.subplots_adjust(hspace=0.7)
pl.savefig("./../Data-Visualization/images/Y-distributions.png",dpi=1000)
```



```
In [29]: import numpy as np
```

```
import scipy.stats as stats
import pylab as pl
import matplotlib.pyplot as plt

pl.subplot(2,1,1)
sort_z_walk = sorted(z_walk)  #sorted
fit = stats.norm.pdf(sort_z_walk, np.mean(sort_z_walk), np.std(sort_z_walk))  #this i
pl.plot(sort_z_walk,fit,'-o',color='black',alpha=0.2)
pl.xlabel('Acceleration(z-axis)')
pl.ylabel('Probability Density')
pl.title('Walking')
pl.hist(sort_z_walk,normed=True,color='skyblue')#use this to draw histogram of your d

pl.subplot(2,1,2)
sort_z_run = sorted(z_run)  #sorted
sort_z_run_np = np.array(sort_z_run)
indices_arr = np.where(np.logical_and(sort_z_run_np>=sort_z_walk[0], sort_z_run_np<=so
first = indices_arr[0][0]
last = indices_arr[0][len(indices_arr[0])-1]
sort_z_run_mod = sort_z_run_np[first:last]
fit = stats.norm.pdf(sort_z_run_mod, np.mean(sort_z_run_mod), np.std(sort_z_run_mod))
pl.plot(sort_z_run_mod,fit,'-o',color='black',alpha=0.2)
pl.hist(sort_z_run_mod,normed=True,color='coral')      #use this to draw histogram of
pl.xlabel('Acceleration(z-axis)')
pl.ylabel('Probability Density')
pl.title('Running')
pl.subplots_adjust(hspace=0.7)
pl.savefig("./../Data-Visualization/images/Z-distributions.png",dpi=1000)
```


Walking
Probability Density
0.0
0.1
0.2
6
8
10
12
14
Acceleration(z-axis)
Running
Probability Density
0.00
0.05
0.10
0.15
6
8
10
12
14
Acceleration(z-axis)