

Data vectors in Python and R

March 5, 2019

1 Code in Python

```
In [1]: import matplotlib.pyplot as plt
import math
```

1.1 Variables

```
In [2]: #All the values from the graph
valx = [0, 11, 10, 8, 5, 8, 6, 6, 9, 4, 5, 4, 4, 4, 8, 8, 9, 3, 8, 4, 5, 3, 10,
        6, 4, 6, 4, 3, 8, 8, 8, 7, 5, 7, 8, 8, 3, 5, 5, 6, 4, 11, 7, 3, 10, 3,
        2, 2, 5, 4, 5, 7, 7, 5, 2, 3, 6, 4, 11, 6, 2, 4, 7, 7, 9, 3, 3, 6, 3, 7,
        2, 7, 5, 7, 6, 4, 10, 5, 4, 6, 6, 6, 10, 8, 9, 7, 2, 6, 5, 9, 2, 7, 8, 5,
        7, 10, 1, 11, 6, 9, 6, 4, 10, 3, 4, 6, 10, 1, 4, 7, 8, 7, 9, 1, 6, 3, 3,
        3, 4, 6, 4, 7, 7, 8, 8, 8, 6, 9, 7, 6, 2, 3, 10, 6, 2, 6, 5, 7, 7, 8, 5,
        4, 5, 4, 4, 8, 4, 4, 6, 4, 5, 3, 7, 7, 11, 15, 8, 5, 7, 1, 7, 4, 7, 6, 9,
        4, 3, 3, 4, 12, 0]

length = len(valx)

#To compute the mean and standard deviation
summ = 0
stand = 0
```

1.2 Compute the mean and the standard deviation

```
In [3]: #Compute the mean
#Is equal to zero if there is no value
for add in valx:
    summ += add
avg = summ/length if summ != 0 else 0

#Compute the standard deviation
for std in valx:
    stand += (std-avg)*(std-avg)
stand = stand/length
stand = math.sqrt(stand)
```

```
#Print the mean and the standard deviation
print(avg, ";", stand)
```

5.83625730994152 ; 2.612850837886284

1.3 Create the graph

In [5]: *#Create the graph*

```
plt.plot(valx, drawstyle = "steps-pre", linewidth = 2)

#Add a green line representing the mean
plt.plot([0, length], [avg, avg], color='g', linestyle='-', linewidth=1)

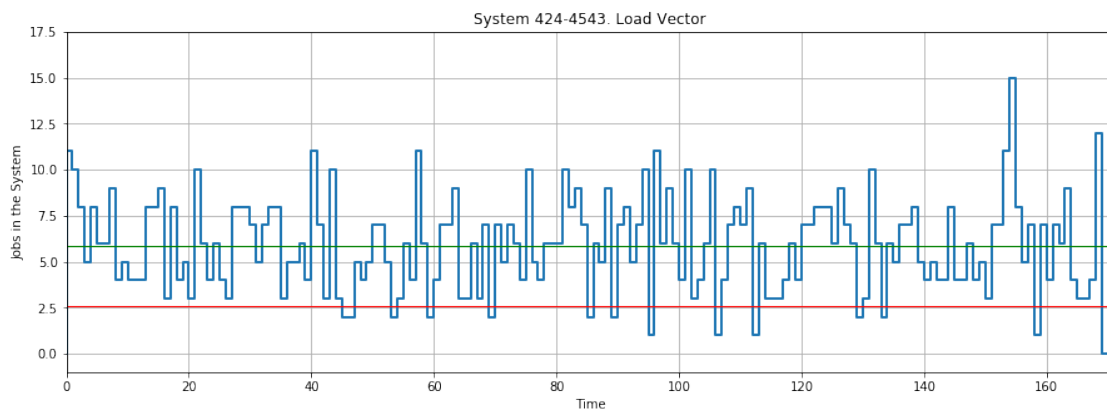
#Add a red line representing the standard deviation
plt.plot([0, length], [stand, stand], color='r', linestyle='-', linewidth=1)

#Set up the axis (not mandatory but better)
axis = plt.gca()
axis.set_xlim([0, length])
axis.set_ylim([-1, 17.5])

#Resize the figure
fig_size = plt.rcParams["figure.figsize"]
fig_size[0] = 15
fig_size[1] = 5

#Add the title and the axis caption
plt.title('System 424-4543. Load Vector')
plt.ylabel('Jobs in the System')
plt.xlabel('Time')

#Add the grid
plt.grid(True)
```



2 Code in R

2.1 Variables

```
In [1]: #All the values from the graph
valy <- c(0, 11, 10, 8, 5, 8, 6, 6, 9, 4, 5, 4, 4, 4, 8, 8, 9, 3, 8, 4, 5, 3, 10,
        6, 4, 6, 4, 3, 8, 8, 8, 7, 5, 7, 8, 8, 3, 5, 5, 6, 4, 11, 7, 3, 10, 3,
        2, 2, 5, 4, 5, 7, 7, 5, 2, 3, 6, 4, 11, 6, 2, 4, 7, 7, 9, 3, 3, 6, 3, 7,
        2, 7, 5, 7, 6, 4, 10, 5, 4, 6, 6, 6, 10, 8, 9, 7, 2, 6, 5, 9, 2, 7, 8, 5,
        7, 10, 1, 11, 6, 9, 6, 4, 10, 3, 4, 6, 10, 1, 4, 7, 8, 7, 9, 1, 6, 3, 3,
        3, 4, 6, 4, 7, 7, 8, 8, 8, 6, 9, 7, 6, 2, 3, 10, 6, 2, 6, 5, 7, 7, 8, 5,
        4, 5, 4, 4, 8, 4, 4, 6, 4, 5, 3, 7, 7, 11, 15, 8, 5, 7, 1, 7, 4, 7, 6, 9,
        4, 3, 3, 4, 12, 0)

len <- length(valy)

#Vector for the X axis
valx <- c(0:(len-1))
```

2.2 Compute mean and standard deviation

```
In [2]: #Compute the mean
avg = mean(valy)

#Compute the standard deviation
stand = sd(valy)

#Print the mean and the standard deviation
cat(avg, ';', stand)
```

5.836257 ; 2.620524

2.3 Create the graph

```
In [4]: #Create the graph
plot(valx, valy, "s", main = "System 424-4543. Load Vector", col = "#318CE7", lwd = 3,

#Add a green line representing the mean
abline(h=avg, col="green")

#Add a red line representing the standard deviation
abline(h=stand, col="red")

#Add the grid
abline(v = 0:173, lty = 2, col = "grey")
abline(h = 0:5, lty = 2, col = "grey")

#Resize the graph
options(repr.plot.width=15, repr.plot.height=8)
```

System 424-4543. Load Vector

