

Session02-04-Matplotlib

April 20, 2021

1 Perform any necessary initialisations

[1]:

[2]: `%matplotlib inline`

Load the xyz.csv file from <https://gist.githubusercontent.com/ariskou/6441e960da38d395b1305e9d388f92d5/raw/6>

[3]:

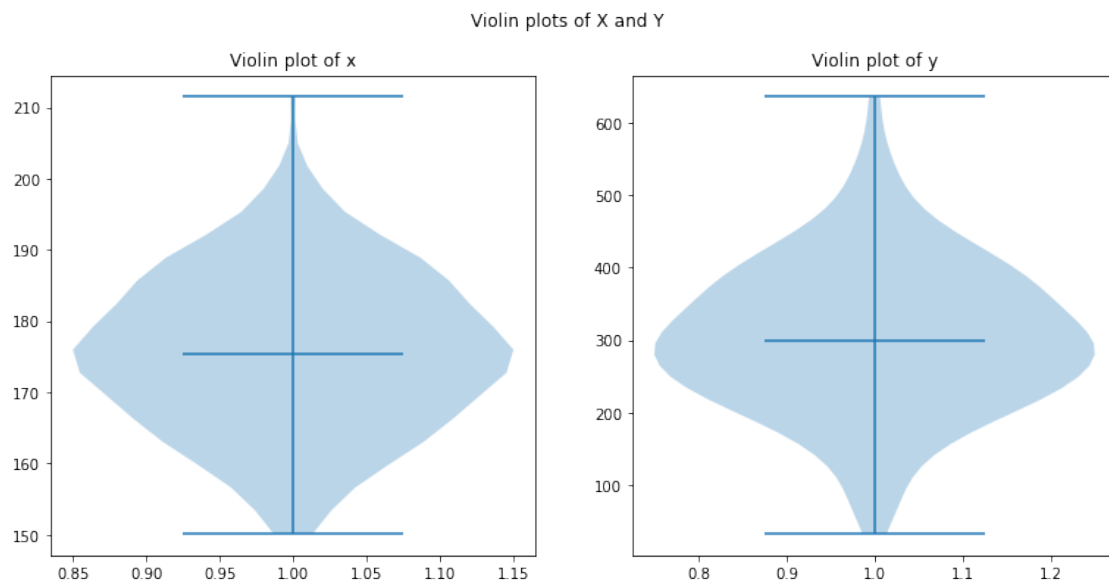
1. Plot “Violin plots” of the distribution of the columns **x** and **y** (i.e., two violin plot, one for each column) in the **same figure**.

Make sure to plot in the violin plots the **median**, **minimum**, and **maximum** but **not the mean/average**.

Set the figure **title** to “Violin plots of X and Y”, and put a title on each of the subplots as “Violin plot of x” and “Violin plot of y”.

[4]:

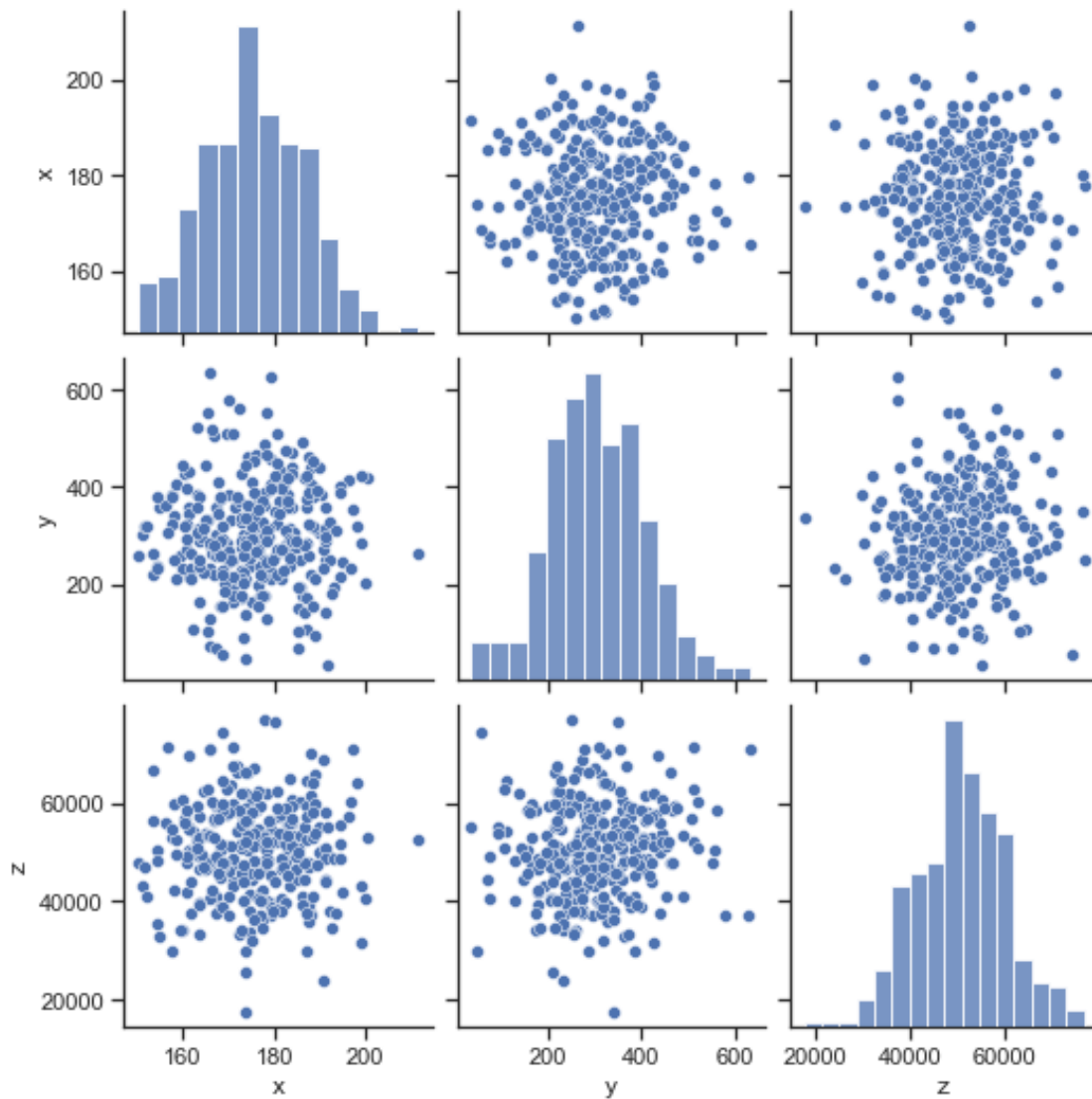
[4]: `Text(0.5, 0.98, 'Violin plots of X and Y')`



2. Plot the “scatter plot matrix” of the columns `x`, `y`, and `z`

[5]:

[5]: <seaborn.axisgrid.PairGrid at 0x655fef92abe0>



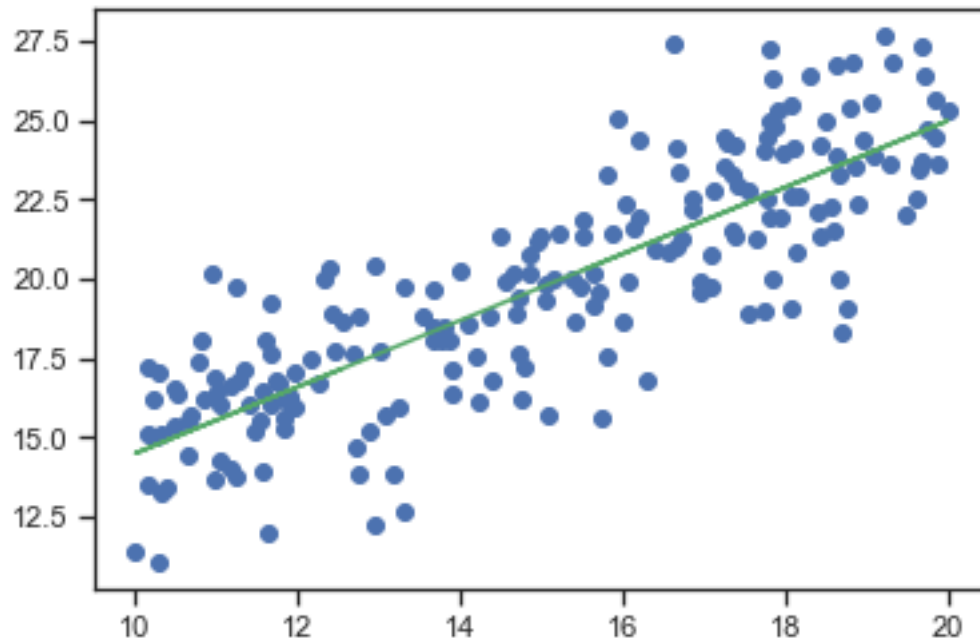
3. You are given the following DataFrame `df_lin`.

```
[6]: x = np.random.uniform(10, 20, 200)
      y = x + 5 + np.random.normal(0, 2, 200)
      df_lin = pd.DataFrame(data=dict(x=x, y=y))
```

3.1. Plot the scatter diagram of the x and y columns and superimpose a linear regression line in green colour.

[7]:

[7]: [<matplotlib.lines.Line2D at 0x655fed599940>]



4. In the following for loop, one random value is generated at every step.

Create a simple line plot, but that **automatically updates** as every new value is generated.

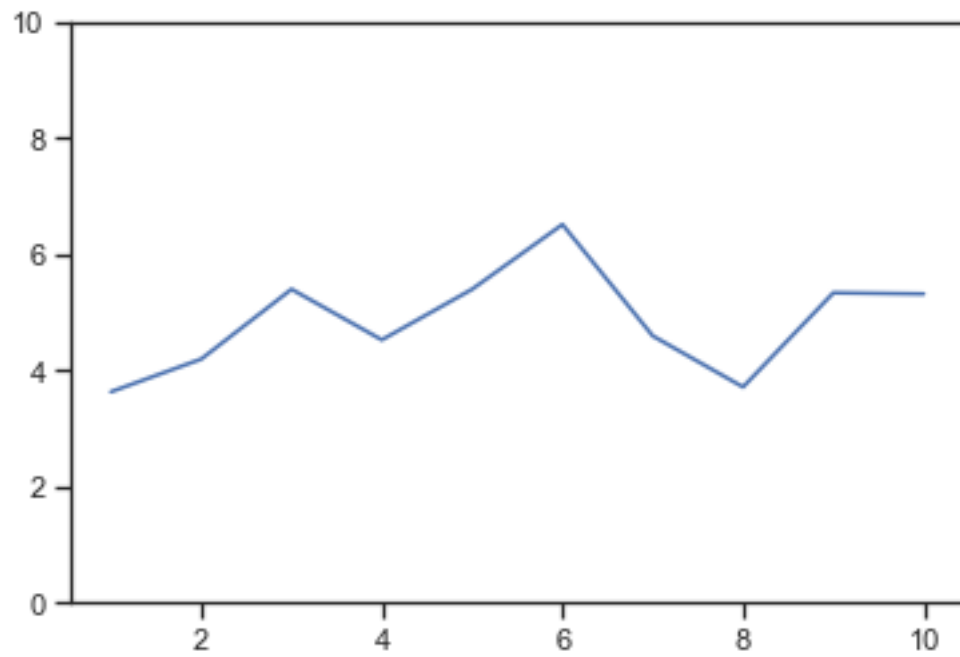
Wait/delay 0.2 seconds between each data point update to create an animation.

Set the minimum of the y axis to 0 and the maximum of the y axis to be equal to 10.

NOTE: There are multiple ways of achieving this and it's a bit tricky, but very useful.

```
[8]: def get_next_point(max_n):  
    for i in range(1, max_n+1):  
        arr = np.random.normal(5, 1, 1)  
        yield i, arr[0]
```

```
[8]: # You can also put thing here also  
for x, y in get_next_point(10):  
    pass  
    # Complete here
```



[]: