

- הכרות עם שפת התכנות פייתון (PYTHON)
- הכרות עם סביבת העבודה של PYTHON
- הכרות עם התכנות הבסיסי בפייתון

➤ ספריות – הכנה ללמידת מכונה:

matplotlib✓

Pandas✓

numpy✓

seaborn✓

הכל נמצא ברשת, צריך רק לדעת איפה לחפש:

אתרים ללמוד עצמי - matplotlib

– <https://matplotlib.org/3.1.1/tutorials/index.html>

– <https://reshetech.co.il/machine-learning-tutorials/matplotlib-the-essentials>

– https://www.w3schools.com/python/matplotlib_intro.asp

הכל נמצא ברשת, צריך רק לדעת איפה לחפש:

אתרים ללמוד עצמי - seaborn

– <https://seaborn.pydata.org/>

– <https://reshetech.co.il/machine-learning-tutorials/simpler-better-faster-plots-with-seaborn>

- ✓ ספריה עזר להצגה ויזואלית של הנתונים – גרפים ותמונות
- ✓ Pyplot תת-מודול שבו נמצאות רוב הפונקציות



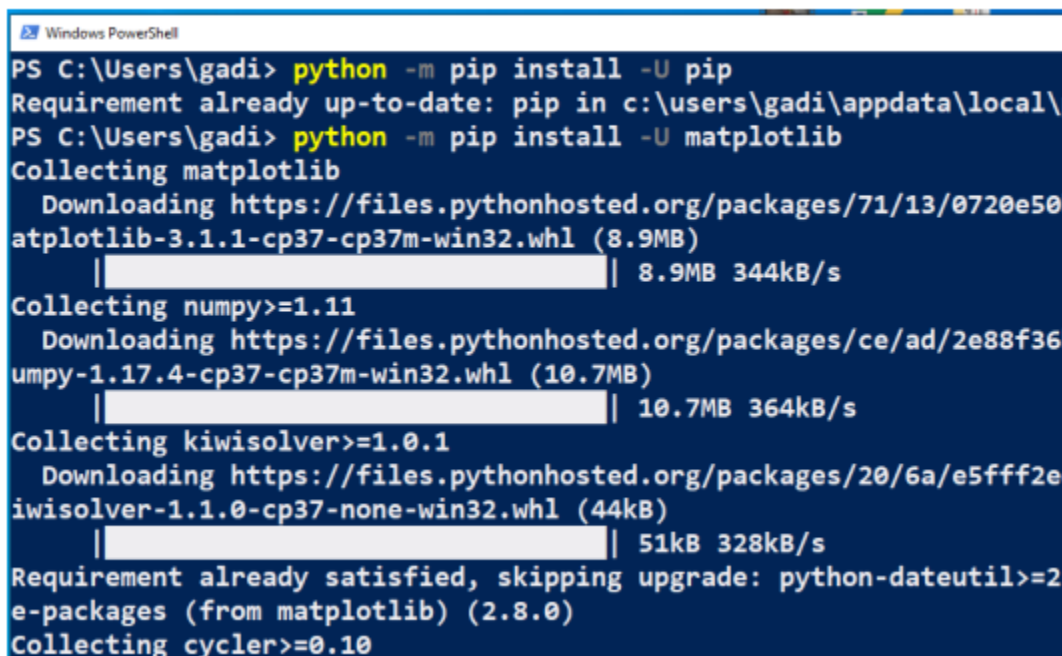
ספריית Matplotlib

- התקנה
- שרטוט גרפים: קו ישר, גרף פיזור, סינוס, גרף פיזור
- עיצוב מערכת צירים – חלוקת הגרף לתתי גרפים
- תיאור גרפי של נתונים מתוך קבצים
- עיצוב מערכת צירים – ראשית צירים (הרחבה)

התקנת הספרייה תתבצע על ידי ההוראה הבאה:

```
> python -m pip install -U pip
> python -m pip install -U matplotlib
> python -m pip install -U Pillow
> python -m pip install -U numpy
```

נקבל כפלט את המסך הבא:



```
Windows PowerShell
PS C:\Users\gadi> python -m pip install -U pip
Requirement already up-to-date: pip in c:\users\gadi\appdata\local\
PS C:\Users\gadi> python -m pip install -U matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/71/13/0720e50
atplotlib-3.1.1-cp37-cp37m-win32.whl (8.9MB)
    | 8.9MB 344kB/s
Collecting numpy>=1.11
  Downloading https://files.pythonhosted.org/packages/ce/ad/2e88f36
umpy-1.17.4-cp37-cp37m-win32.whl (10.7MB)
    | 10.7MB 364kB/s
Collecting kiwisolver>=1.0.1
  Downloading https://files.pythonhosted.org/packages/20/6a/e5fff2e
iwisolver-1.1.0-cp37-none-win32.whl (44kB)
    | 51kB 328kB/s
Requirement already satisfied, skipping upgrade: python-dateutil>=2
e-packages (from matplotlib) (2.8.0)
Collecting cycpler>=0.10
```

```
%matplotlib inline  
import matplotlib.pyplot as plt
```



מלבד matplotlib כדאי לייבא את:

```
% matplotlib inline
```

שמאפשרת לצייר את התרשימים בתוך ה-jupyter

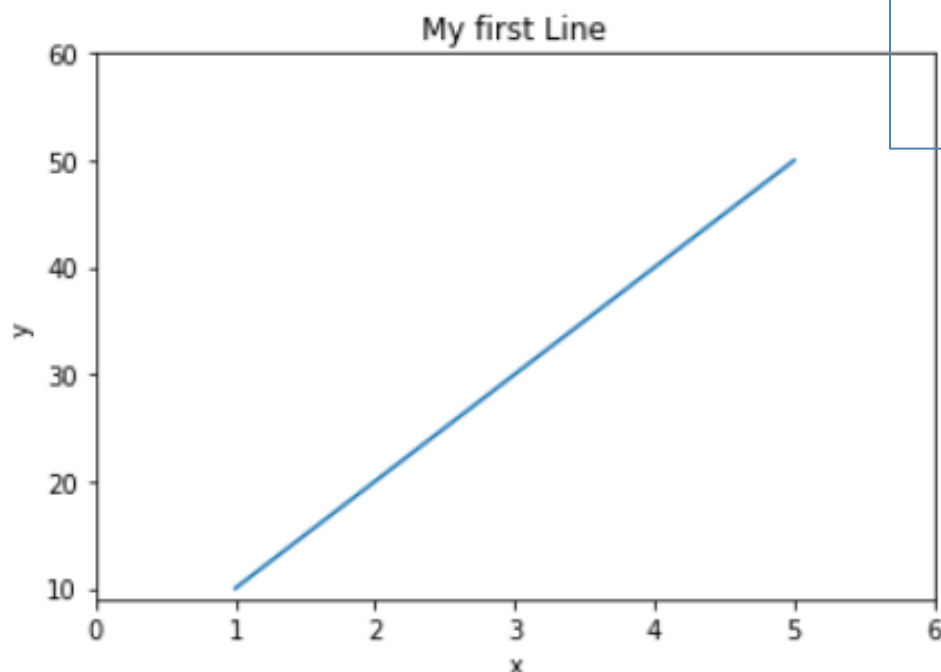
notebook ללא פקודה מיוחדת

אם לא מייבאים אותה יש להקפיד להוסיף את הפקודה:

```
plt.show()
```



```
x=[1,5]
y=[10,50]
plt.plot(x,y)
plt.title('My first Line')
plt.axis([0, 6, 9, 60])
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



הפעולה plot

קובעת מהם הנתונים שיש להציג על הגרף.

הפעולה title

קובעת את הכותרת של הגרף.

הפעולה axis

קובעת את טווח הערכים של מערכות הצירים

הפעולה xlabel- ylabel

קובעים את הכותרות לצירים X- Y בהתאמה.

הפעולה show

מחוללת את הגרף.



חשוב להקפיד על:

- כותרת הגרף
- כותרות לצירים
- טווח מערכת הצירים

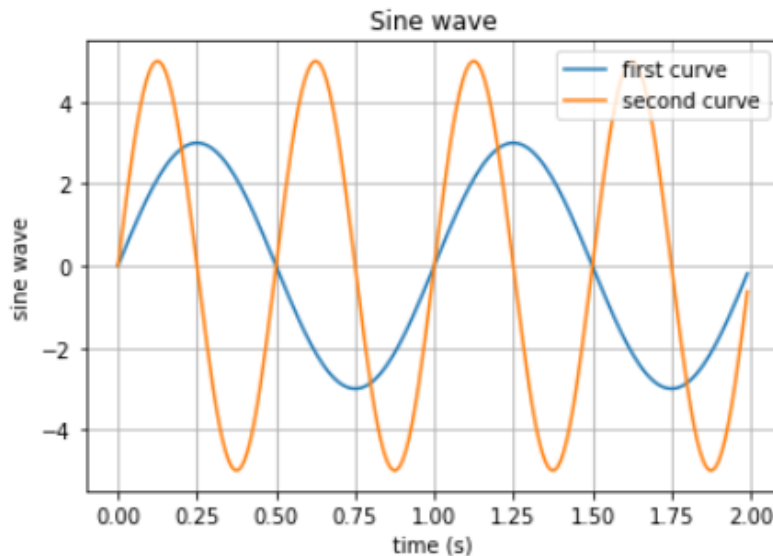


```
t = np.arange(0.0, 2.0, 0.01)
y1 = 3*np.sin(2 * np.pi * t)
y2 = 5*np.sin(2 * np.pi * 2* t)

plt.plot(t, y1, label='first curve')
plt.plot(t, y2, label='second curve')

plt.grid() # display grid lines
plt.title("Sine wave")
plt.xlabel("time (s)")
plt.ylabel("sine wave")

plt.legend()
plt.show()
```



lable

plt.legend()

מוסיפה לגרף מקרא

grid

מוסיפה רשת קווים פנימית

לפי המשוואות הבאות:

$$y1 = \text{CONST1} * \sin(\text{CONST1} * \pi * \text{arr1})$$

$$y2 = \text{CONST2} * \sin(\text{CONST2} * \pi * \text{arr1})$$

CONST1 - מספר כלשהו

CONST2 - מספר כלשהו

arr1 - מערך שמתחיל מ 0 עד 2 בקפיצות של 0.01

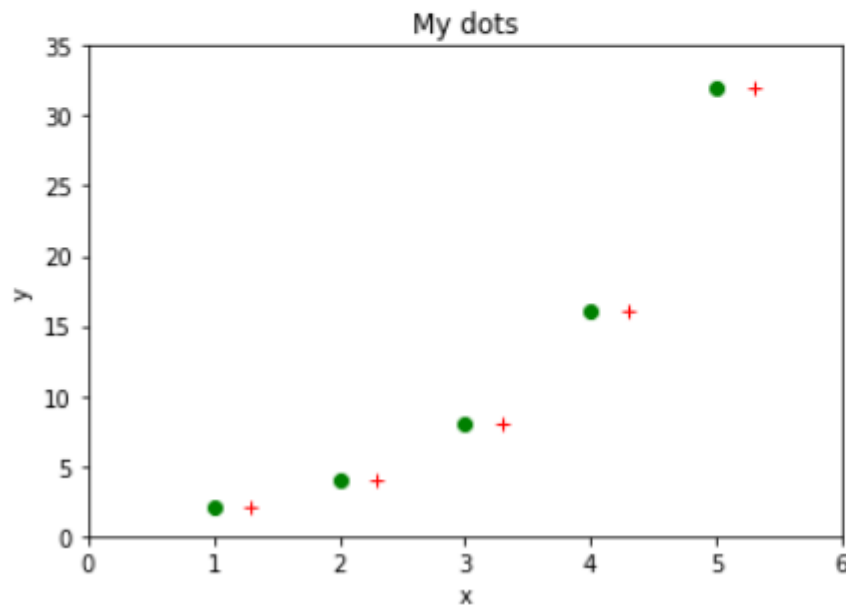
יש להקפיד על:

כותרות, טווח מערכת צירים, קווי רשת פנימיים, ומקרא

פיזור נקודות על גבי מערכות צירים – שיטה 1

✓
0s

```
plt.plot([1, 2, 3, 4, 5], [2, 4, 8, 16, 32], 'go') #g=green color o=הנקודה של הצורה  
plt.plot([1.3, 2.3, 3.3, 4.3, 5.3], [2, 4, 8, 16, 32], 'r+') #r=red color +=הנקודה של הצורה  
plt.axis([0, 6, 0, 35])  
plt.title('My dots')  
plt.xlabel('x')  
plt.ylabel('y')  
plt.show()
```



פרמטר ראשון הוא ציר ה-x

פרמטר שני ציר הוא ציר ה-y

'g' – צבע ירוק

'o' – הופך את התרשים לתרשים פיזור

עם נקודות

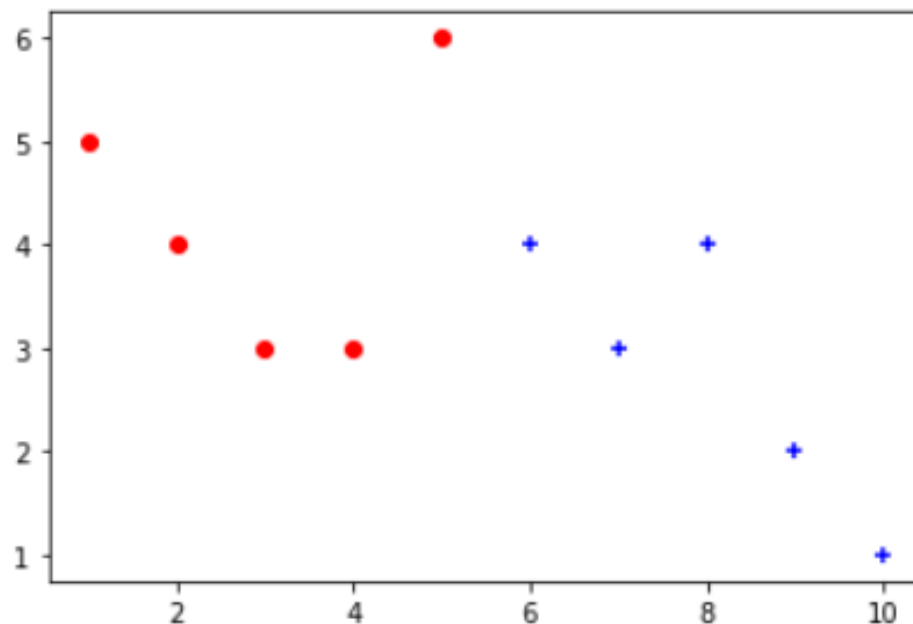
'r' – צבע אדום

'+' – הופך את התרשים לתרשים פיזור

עם +

גרף פיזור נקודות בצבעים שונים – שיטה 2

```
x1 = [1,2,3,4,5]
y1 = [5,4,3,3,6]
x2 = [6,7,8,9,10]
y2 = [4,3,4,2,1]
plt.scatter(x1, y1, color='red') # נקודות בצבע אדום
plt.scatter(x2, y2, c='blue', marker="+") # פלוסים בצבע כחול
plt.show()
```



scatter

גרף פיזור

צבע=c או צבע= color

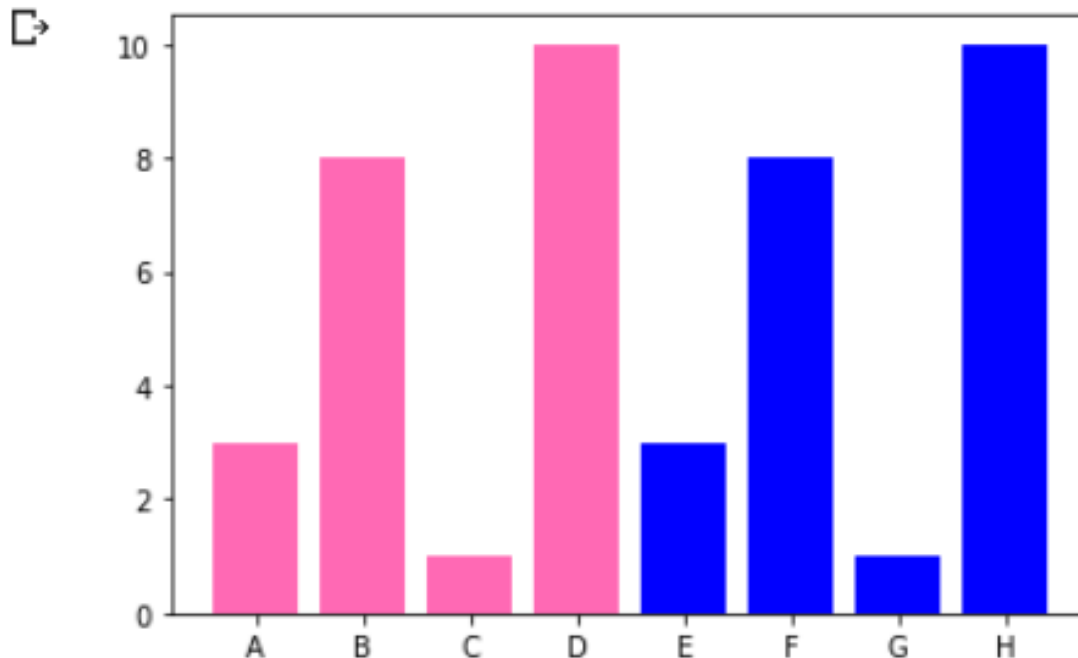
מסמנת את הנקודה בצבע הרצוי

marker

משנה את צורת הסימון

```
▶ x1 = np.array(["A", "B", "C", "D"])  
x2 = np.array(["E", "F", "G", "H"])  
y = np.array([3, 8, 1, 10])  
plt.bar(x1,y, color = "hotpink")  
plt.bar(x2,y, color = "blue")  
plt.show()
```

bar
גרף עמודות
צבע= color
מסמנת את העמודה בצבע הרצוי



יש לשנות את קטע הקוד הקודם כך ש:

עמודה E תהיה בגובה 5

עמודה F תהיה בגובה 4

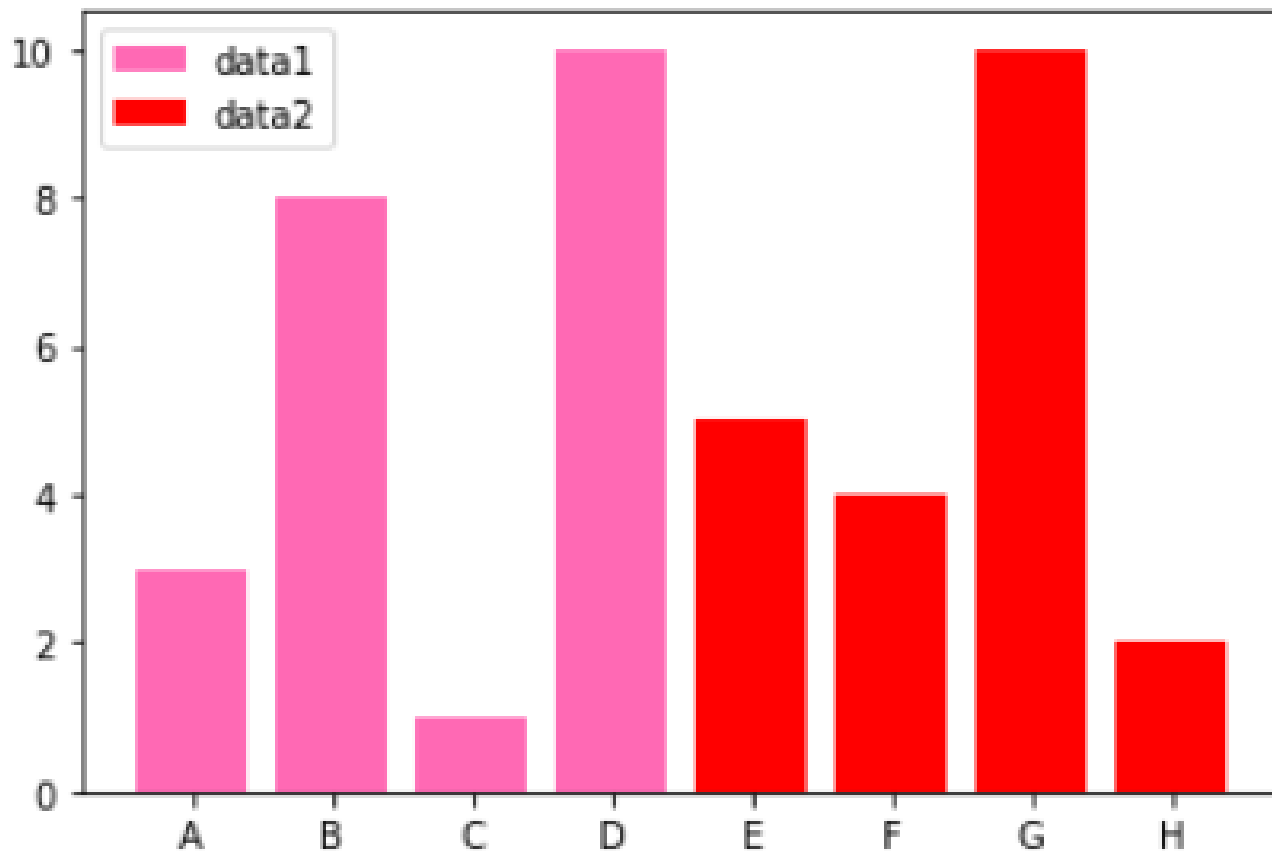
עמודה G תהיה בגובה 10

עמודה H תהיה בגובה 2

עמודות D-A ללא שינוי

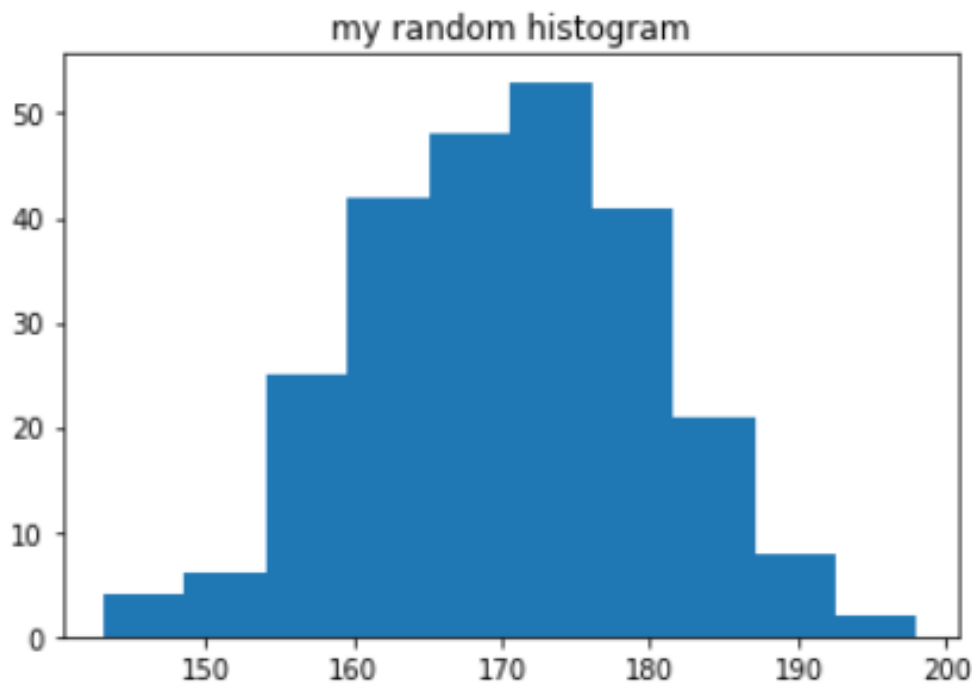
יש להוסיף מקרא לעמודות

תרגול 2 - גרף עמודות פתרון





```
x = np.random.normal(170, 10, 250)
#We use NumPy to randomly generate an array with 250 values,
#where the values will concentrate around 170, and the standard deviation is 10
plt.title('my random histogram')
plt.hist(x)
plt.show()
```



hist

גרף הסטנוגרמה

שימושי למציאת שכיחויות

הצגה של מספר גרפים יחד באותה תמונה – דרך 1



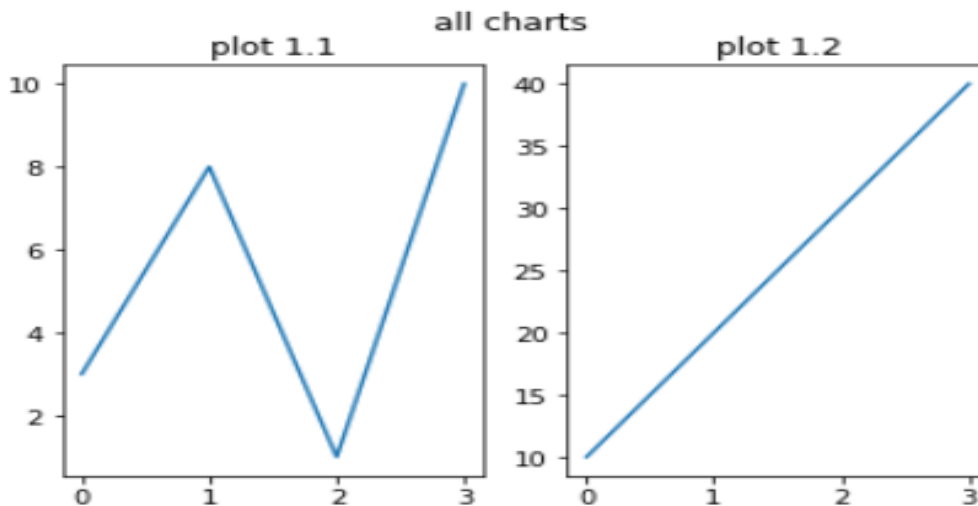
```
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("plot 1.1")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("plot 1.2")

plt.suptitle("all charts")
plt.show()
```

`plt.subplot(row,col,position)`
קביעת מיקום בגרף

`plt.suptitle("all charts")`
כותרת כל הגרפים



הצגה של מספר גרפים יחד באותה תמונה – דרך 2

```
rows=1  
cols=2  
fig, (ax1, ax2) = plt.subplots(rows, cols)
```

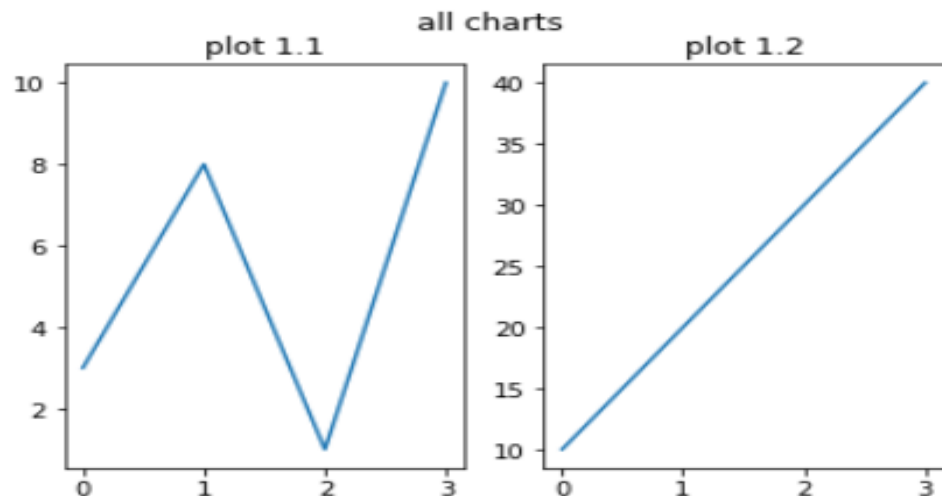
```
#plot 1:  
ax1.set_title("plot 1.1")  
x = np.array([0, 1, 2, 3])  
y = np.array([3, 8, 1, 10])  
ax1.plot(x,y)  
  
#plot 2:  
ax2.set_title("plot 1.2")  
x = np.array([0, 1, 2, 3])  
y = np.array([10, 20, 30, 40])  
ax2.plot(x,y)
```

```
fig.suptitle("all charts")
```

`fig, (ax1, ax2) = plt.subplots(rows, cols)`
יצירת שני אובייקטים לשני הגרפים

`fig.suptitle("all charts")`
כותרת כל הגרפים

```
Text(0.5, 0.98, 'all charts')
```



□ יש להציג בתמונה אחת 4 גרפים באופן הבא:

□ בגרף הימני למעלה תציגו גרף סינוס

□ בגרף הימני למטה תציגו גרף הסטוגרמה

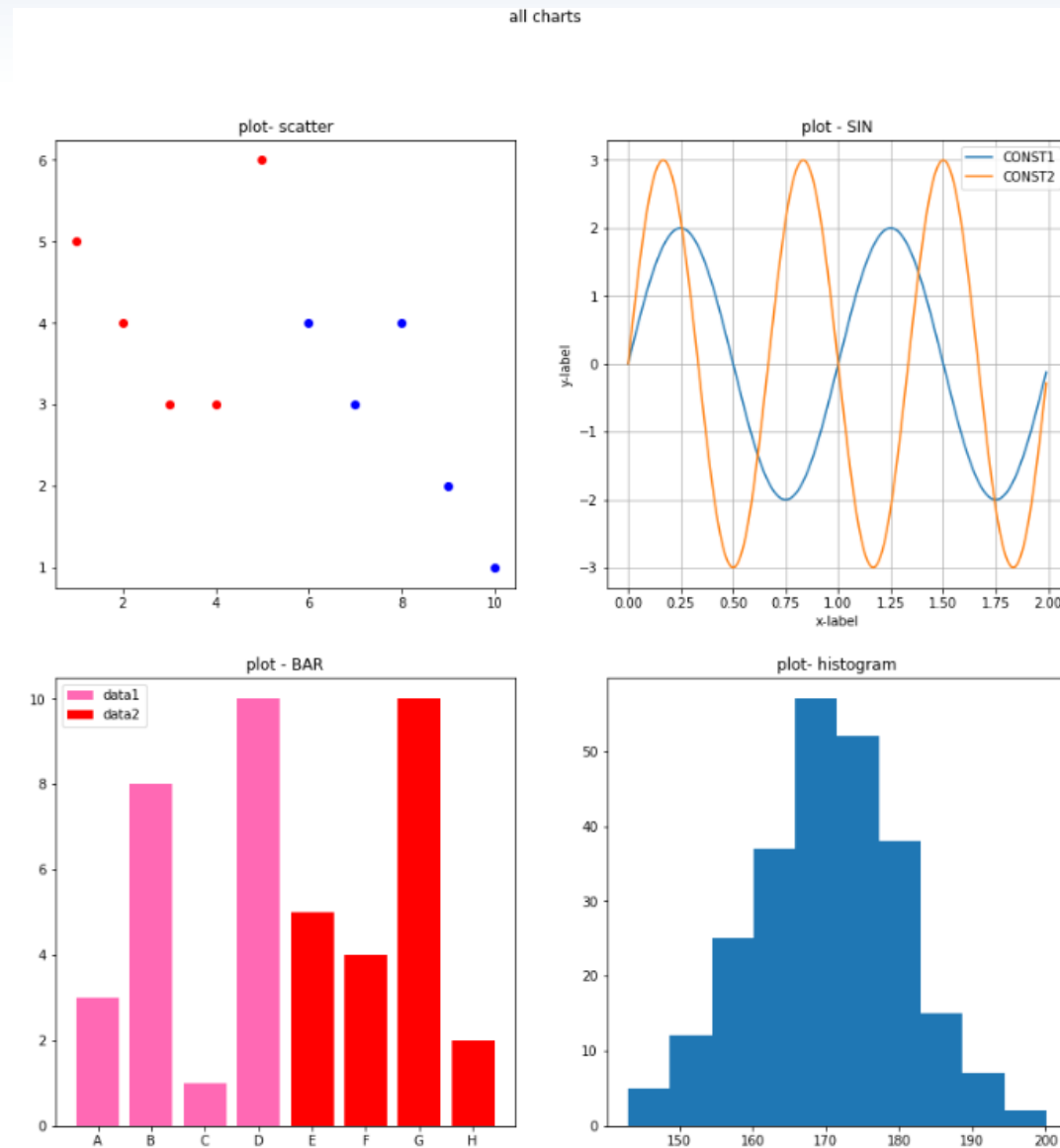
□ בגרף השמאלי למעלה תציגו גרף פיזור

□ בגרף השמאלי למטה תציגו גרף עמודות

□ תנו כותרת מתאימה לכל גרף

□ תנו כותרת מתאימה לתמונה כולה

תרגול 3 - 4 גרפים בתמונה אחת – תוצאה צפויה

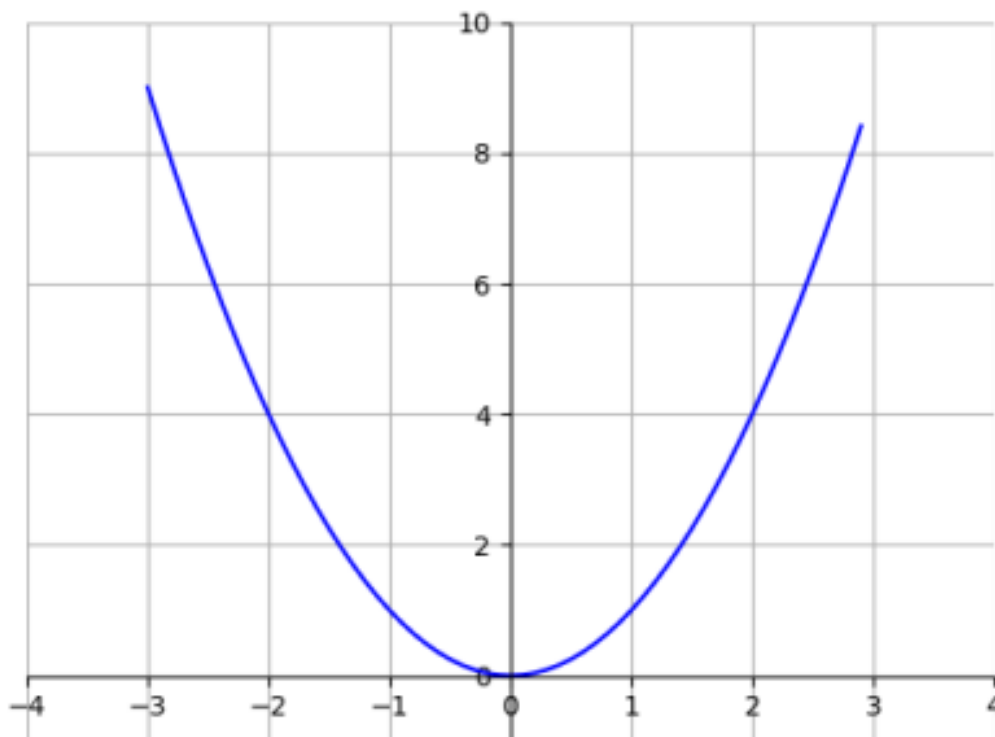


הרחבה – יצירת מערכת צירים מותאמת לגרף

נבחן גרף של הפונקציה הפולינומית הבא:

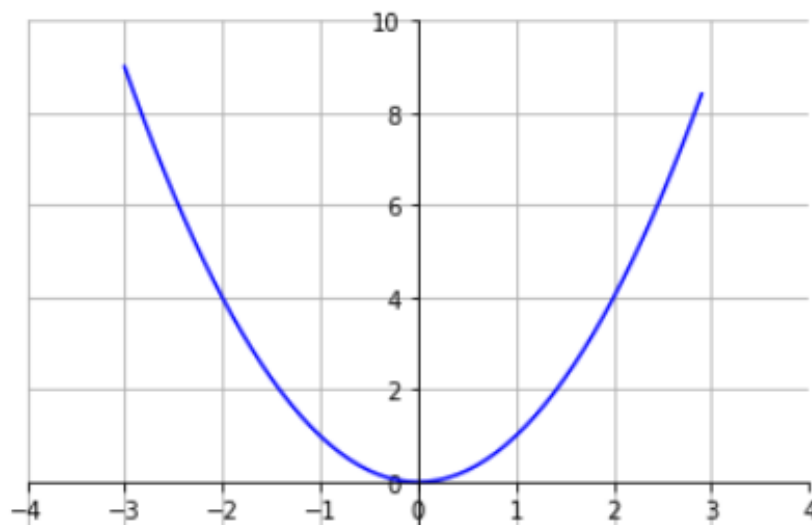
$$y = x^2$$

נבקש ליצור מערכת צירים מותאמת לגרף באופן הבא:



הרחבה – יצירת מערכת צירים מותאמת לגרף - קוד

```
▶ ax = plt.gca()
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['top'].set_color('none')
plt.xlim([-4, 4])
plt.ylim([-1, 10])
plt.grid()
x = np.arange(-3.0, 3.0, 0.1)
plt.plot(x, x**2, color = "b")
plt.show()
```



✓ שרטוט קו ישר, באמצעות ערכי X וערכי Y הניתנים כמערכי

Numpy

✓ שינוי סוג הקו, הוספת מרקרים, שינוי גודל וצבע הגרף בכל

אחד מסוגי הגרפים

✓ כותרות לצירים, כותרת לגרף, הוספת קווי רשת

✓ הצגת תתי גרפים באמצעות subplots

✓ סוגי גרפים – קו ישר, פיזור, עמודות, הסטוגרמה