

יסודות

פעילות 1 - תכנות גרפים תוך שימוש ב- Matplotlib

Matplotlib היא ספרייה Python המייצרת גרפים דו-מימד במגוון פורמטים. הספרייה מספקת ממשק תכנותי הדומה ל-MATLAB. בעזרת הספרייה ניתן ליצור בפשטות וביעילות היסטוגרמות וגרפים שונים כמו עמודות ומערכות צירים X,Y. כמו כן Matplotlib מאפשרת הצגת תמונות ונתונים כקובץ תמונה. במדריך זה נעשה שימוש משמעותי ב-Matplotlib כדי להבין תהליכים המתבצעים במערכות למידת מכונה שנפתח בהמשך המדריך.

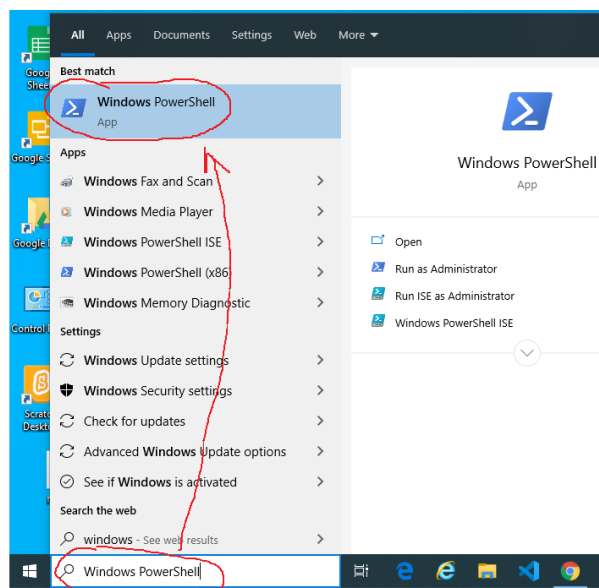
מקורות:

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

<https://www.science-emergence.com/Articles/How-to-create-a-scatter-plot-with-several-colors-in-matplotlib/>

התקנת הספרייה

נפתח את חלון הפקודות Windows PowerShell על ידי ההוראה הבא:



נבדוק תחילה שהמפרש של Python מותקן במחשב, נעשה זאת על ידי ההוראה:

```
> python --version
```

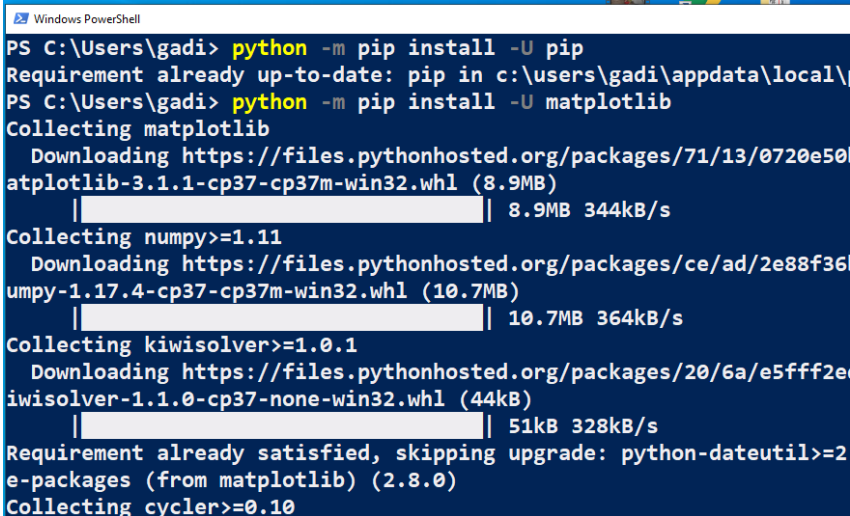
```
Windows PowerShell
PS C:\Users\gadi> python --version
Python 3.7.4
PS C:\Users\gadi>
```

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
```

** בפעילות זו נעשה שימוש גם בספריות Pillow ו-numpy שנלמד בהמשך המדריך.

נקבל כפלט את המסך הבא:



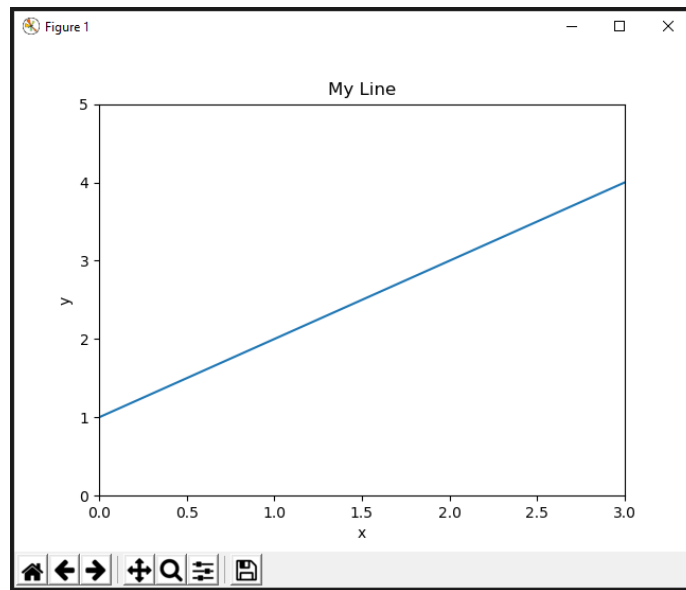
```
Windows PowerShell
PS C:\Users\gadi> python -m pip install -U pip
Requirement already up-to-date: pip in c:\users\gadi\appdata\local\
PS C:\Users\gadi> python -m pip install -U matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/71/13/0720e50
atplotlib-3.1.1-cp37-cp37m-win32.whl (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
iwsolver-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 cycler>=0.10
```

משימה 1: יצירת קו ישר על גבי מערכת צירים

נכתוב את הקוד הבא:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.title('My Line')
plt.axis([0, 3, 0, 5])
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

נקבל את הפלט הבא:



מקרא הוראות:

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 לפי הפירוט הבא: `[xmin, xmax, ymin, ymax]`

הפעולה `xlabel` ו- `ylabel` קובעים את הכותרות לצירים X ו- Y בהתאמה.

הפעולה `show` מחוללת את הגרף.

כמובן לא כל הפעולות חיוניות ליצירת הגרף, להלן הגרסה המינימלית ליצירת קו.

```
import matplotlib.pyplot as plt
plt.plot([1, 4])
plt.show()
```

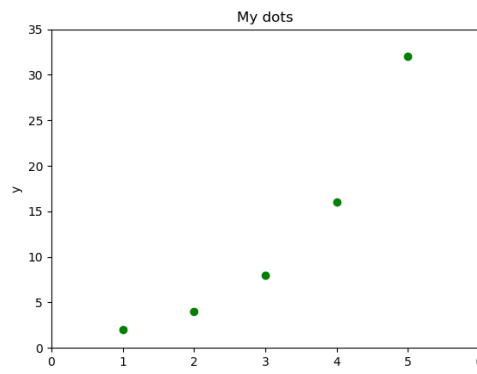
משימה 2: פיזור נקודות על גבי מערכות צירים

נכתוב את הקוד הבא:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4, 5], [2, 4, 8, 16, 32], 'go')
plt.axis([0, 6, 0, 35])
plt.title('My dots')
plt.xlabel('x')
plt.ylabel('y')
```

```
plt.show()
```

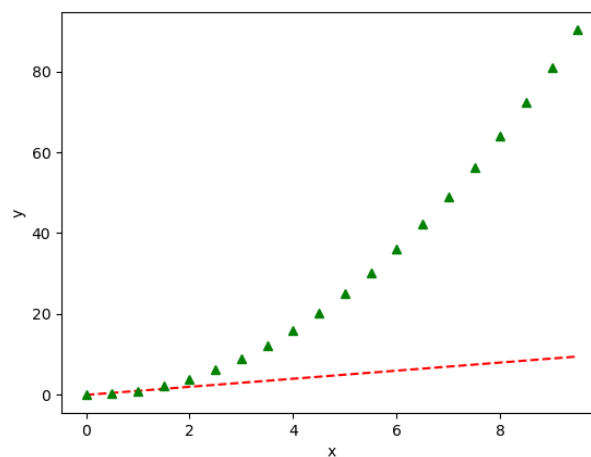
*הכיתוב go מציין שמדובר על נקודות בצורת עיגול 'o' בצבע ירוק 'g'



כמובן ניתן לשרטט מספר מערכות צירים בצבעים שונים להלן דוגמה:

```
import numpy as np
import matplotlib.pyplot as plt
t = np.arange(0.0, 10.0, 0.5)
print(t)
plt.plot(t, t, 'r--')
plt.plot(t, t**2, 'g^')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

נקבל את הפלט הבא:



בקוד זה ובבאים אחריו נעשה שימוש בפעולות השייכות לספרייה המתמטית numpy של Python.

הפעולה `np.arange` מייצרת סדרה של מספרים מהערך הראשון שהפעולה מקבלת עד לערך האחרון בקפיצות של הערך השלישי שהפעולה מקבלת.

לדוגמה:

```
import numpy as np
t = np.arange(0.0, 3.0, 0.1)
print(t)
```

הפעולה תחזיר את הרשימה הבא:

```
[0. 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9]
```

משימה 3: שרטוט אות סינוס

משוואת את סינוס נתונה להלן:

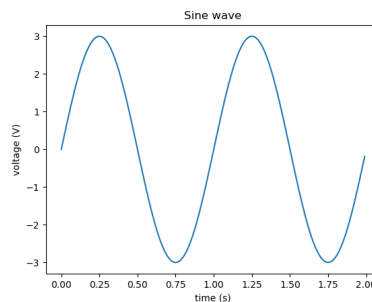
$$y(t) = A \cdot \sin(2\pi ft)$$

נכתוב את הקוד המשתמש בנוסחה הנ"ל כדי לשרטט גרף של אות סינוס:

```
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0.0, 2.0, 0.01)
s = 3*np.sin(2 * np.pi * t)
plt.plot(t, s)
plt.title("Sine wave")
plt.xlabel("time (s)")
plt.ylabel("voltage (V)")
plt.show()
```

נקבל את הפלט הבא:

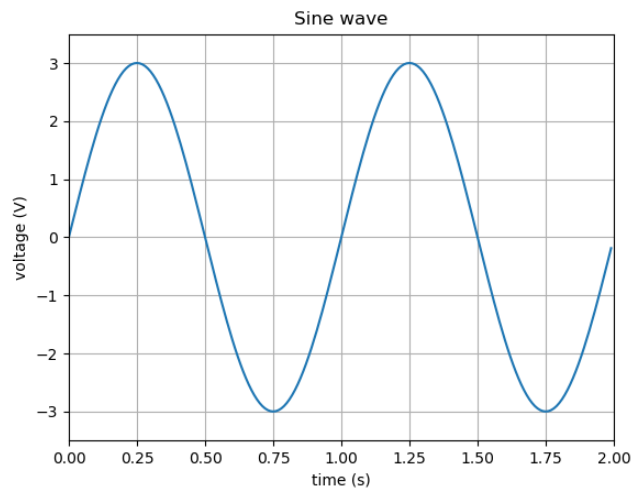


ניתן לשפר את מראה האות על ידי הוספת רשת קווים פנימית (grid) כמו כן קביעת הערכים של מערכת הצירים. נדגים זאת:

```
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(0.0, 2.0, 0.01)
s = 3*np.sin(2 * np.pi * t)
plt.plot(t, s)
plt.axis([0, 2, -3.5, +3.5])
plt.grid()
plt.title("Sine wave")
plt.xlabel("time (s)")
plt.ylabel("voltage (V)")
plt.show()
```

נקבל את הפלט הבא:



תרגיל:

שרטט את האותות הבאים:

$$y(t) = 5 + 5 \cdot \sin(2\pi 500t) \quad .1$$

$$y(t) = 5 \cdot \sin(2\pi 1000t) \quad .2$$

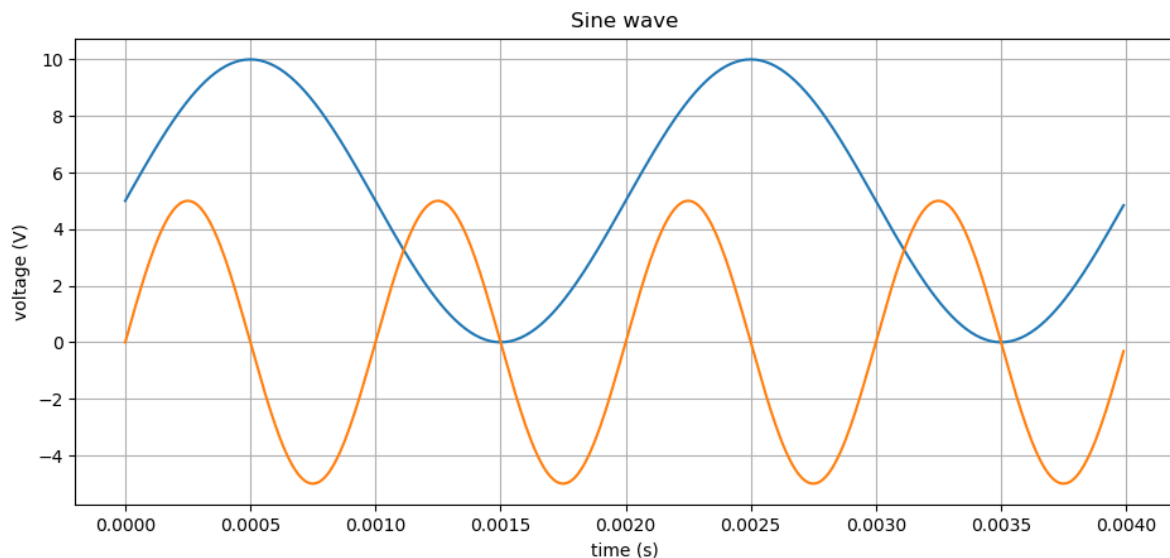
פתרון:

```
import numpy as np
import matplotlib.pyplot as plt
```

```

t = np.arange(0.0, 0.004, 0.00001)
y1 = 5 + 5*np.sin(2 * np.pi * 500* t)
y2 = 5*np.sin(2 * np.pi * 1000* t)
plt.plot(t, y1)
plt.plot(t, y2)
plt.grid()
plt.title("Sine wave")
plt.xlabel("time (s)")
plt.ylabel("voltage (V)")
plt.show()

```



שרטט את האותות הבאים:

$$y = 10x + 6 \quad .1$$

$$y = 2x^2 + 2x - 100 \quad .2$$

פתרון:

```

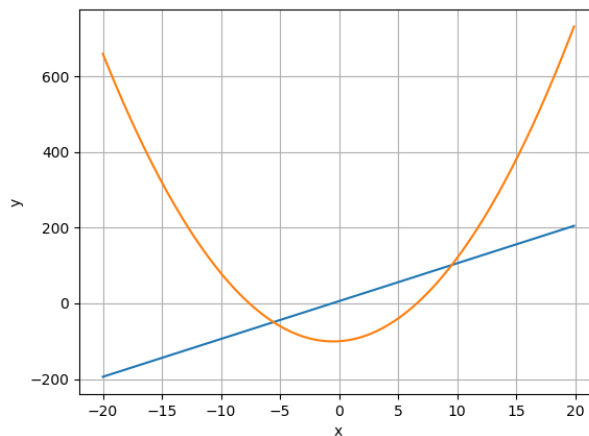
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-20, 20, 0.1)
y1 = 10*x + 6
y2 = 2*x**2+2*x-100

```

```
plt.plot(x, y1)
plt.plot(x, y2)
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

נקבל את הפלט הבא:



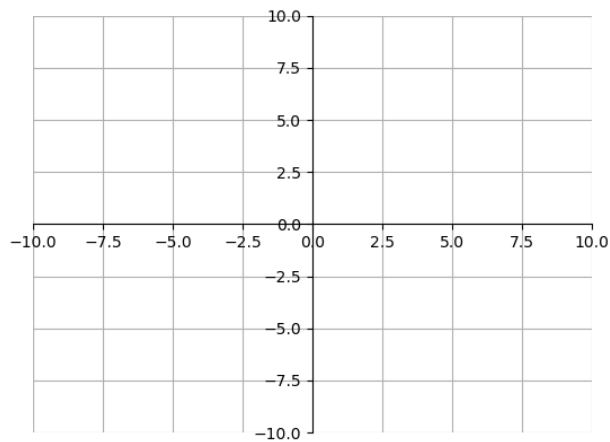
משימה 4: עיצוב מערכת הצירים

נכתוב את הקוד הבא:

```
import matplotlib.pyplot as plt

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([-10, 10])
plt.ylim([-10, 10])
plt.grid()
plt.show()
```


נקבל את הפלט הבא:



משימה 5: שרטוט מספר גרפים על גבי מערכת צירים אחת

נכתוב את הקוד הבא:

```
import numpy as np
import matplotlib.pyplot as plt

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([-10, 10])
plt.ylim([-10, 10])
plt.grid()

x = np.arange(-10,10,0.1)          #x = np.linspace(-10,10,10)
y1 = 0.5*x+2.5
y2 = x+2.5
y3 = 2*x+2.5
plt.plot(x, y1, '-r', label='y=0.5x+2.5')
plt.plot(x, y2, '-g', label='y=x+2.5')
plt.plot(x, y3, '-b', label='y=2x+2.5')
```

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

ההבדל בין הפעולות `arange` ל- `linspace`:

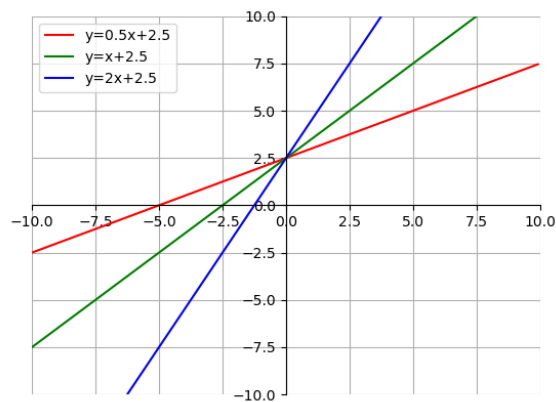
```
x = np.arange(-10,10,0.1)
```

הפעולה `arange` מייצרת רשימה של ערכים מהערך מינוס עשר עד עשר בקפיצות של 0.1. גם הפעולה `linspace` מייצרת סדרה של ערכים באותו הטווח אך הפעם מספר האיברים ברשימה נקבע לפי הפרמטר השלישי, כלומר 10 ערכים.

```
x = np.linspace(-10,10,10)
```

בהמשך המדריך נקדיש פרק שלם לעבודה על מערך הרשימה של `numpy`. בשלב זה קחו בחשבון שהמשתנה `x` הוא רשימה ולא ערך בודד!

לאחר הרצת הקוד נקבל את הפלט הבא:

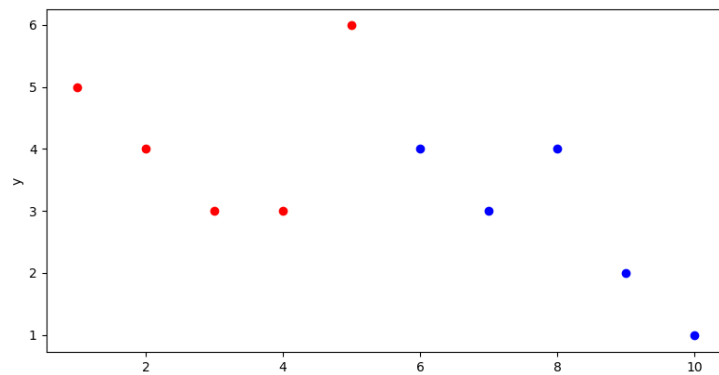


משימה 6: פיזור נקודות בצבעים שונים על גבי מערכות צירים דו-מימד

נכתוב את הקוד הבא:

```
import matplotlib.pyplot as plt
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, c='red')
plt.scatter(x2, y2, c='blue')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

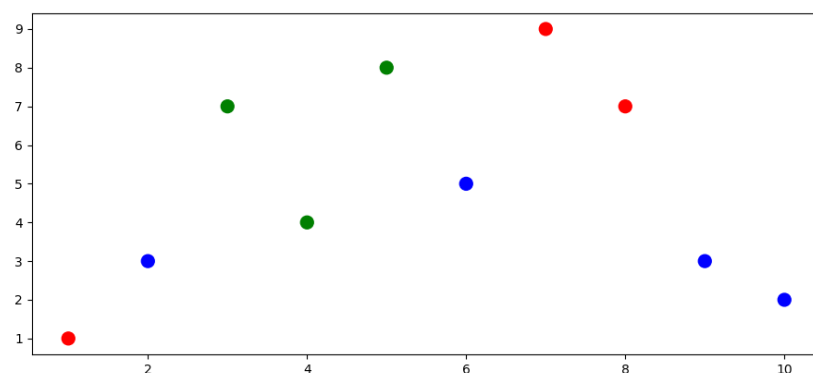
נקבל את הפלט הבא:



ניתן לשפר את הגרף הקודם בכך שנגדיר מערך ייעודי לצבעים של על אחד מהנקודות. כלומר מערך צבעים שממנו כל נקודה תקבל את הצבע שלה. נדגים זאת:

```
import matplotlib.pyplot as plt
import numpy as np
a = np.array([ [1,2,3,4,5,6,7,8,9,10],
               [1,3,7,4,8,5,9,7,3,2] ])
categories = np.array([0,2,1,1,1,2,0,0,2,2])
colormap = np.array(['r', 'g', 'b'])
plt.scatter(a[0], a[1], s=100, c=colormap[categories])
plt.show()
```

נקבל את הפלט הבא:



משימה 7: פיזור נקודות בצבעים שונים על גבי מערכות צירים שלוש מימדים

נכתוב את הקוד הבא:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```

import numpy as np

data = np.array( [ [ 6.0 , 7.0, 5.0],
                  [ 2.0 , 3.0, 7.0],
                  [ 3.0 , 7.0, 2.0],
                  [ 4.0 , 4.0, 8.0],
                  [ 5.0 , 8.0, 9.0],
                  [ 6.0 , 5.0, 7.0],
                  [ 7.0 , 9.0, 4.0],
                  [ 8.0 , 5.0, 1.0],
                  [ 8.0 , 2.0, 3.0],
                  [10.0 , 2.0, 5.0]  ])

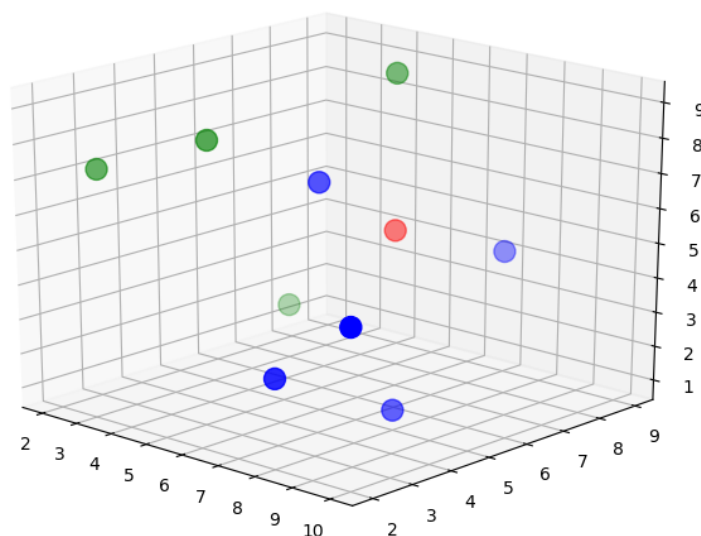
categories = np.array([0,1,1,1,1,1,2,2,2,2,2])
colormap = np.array(['r', 'g', 'b'])

fig = plt.figure()
ax = Axes3D(fig)

ax.scatter(data[:,0], data[:,1],data[:,2], s=150, c=colormap[categories])
plt.show()

```

נקבל את הפלט הבא:

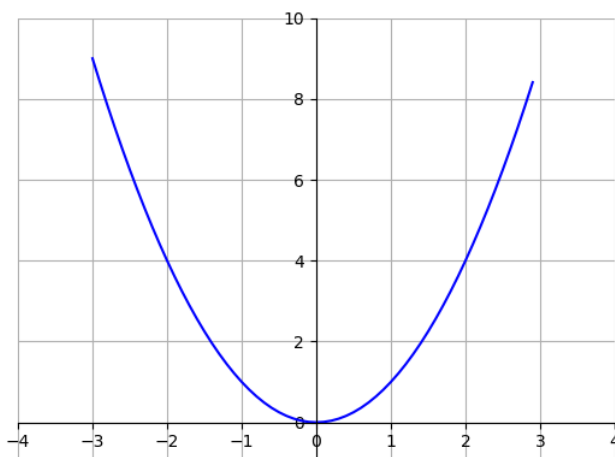


משימה 8: עיצוב מערכת הצירים

נבחן גרף של הפונקציה הפולינומית הבא:

$$y = x^2$$

נבקש ליצור מערכת צירים מותאמת לגרף באופן הבא:



להלן תוכנה המחולל את הגרף הבא:

```
import numpy as np
import matplotlib.pyplot as plt

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()
```

תנאי השימוש

תנאי השימוש במסמך זה הם לפי הסטנדרט הבא:

You are free:

to Share – to copy, distribute and transmit the material
to Remix – to adapt the material

Under the following conditions:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.