

תרגיל בית 2 - תכנות מתקדם

נושא: הספריות Numpy, Matplotlib

הנחיות:

1. נא לכתוב קוד מסודר, תוך שימוש בשמות משתנים בעלי משמעות והערות היכן שנדרש.
2. בתחילת כל מענה, נא להוסיף כהערה את שמות המגישים והאם התייעצתם / נעזרתם בסטודנטים נוספים.

הגשה:

1. יש להגיש את העבודות בזוגות (עפ"י הקבוצות שנרשמתם).
2. יש להגיש מחברת בודדת (Jupyter notebook) עם המטלה, או לחילופין מספר קבצי python בהתאם לשאלות ולאחד אותם לקובץ הגשה בודד (לדוג' בעזרת zip - נא לא לדחוס סביבות וירטואליות תוך כדי!).
3. שם הקובץ המוגש צריך להיות: student1_id_student2_id.ipynb (או zip).

בהצלחה !

15 נק' חלק א' - יצירת נתונים

לטובת מחקר חדש בתחום הסוכרת, דוגמים 16 נבדקים. בכל יום, מבצעים בדיקת רמת סוכר עבור כל אחד מן הנבדקים. הניסוי מתרחש לאורך 21 ימים.

1. (2 נק') - צרו את המשתנה `sugar_range` - מספר **אקראי** (שלם) בטווח שבין 0 ל 10. משתנה זה מייצג את רמת הסוכר המקסימלית שנמדדה.

2. (3 נק') - צרו מטריצה בגודל 16×21 , אשר כל שורה i מייצגת את רמת הסוכר של הנבדק ה- i לאורך 21 ימים. ערכי המטריצה הם מספרים **אקראיים** בטווח שבין 0 ל `sugar_range`. למשל, אם בסעיף 1 הוגרל המספר 5, אז הטווח של ערכי המטריצה הוא בין 0 ל 5.

1	...	3	נבדק ראשון לאורך 21 ימים
:	.	:	
4	...	1	נבדק אחרון לאורך 21 ימים

3. (5 נק') - כתבו את הפונקציה `get_sugar_avg` אשר מקבלת כקלט מטריצה ומחזירה מערך בגודל 16 (מספר הנבדקים) המכיל את ממוצע הסוכר של כל נבדק לאורך 21 הימים. למשל, האיבר ה-0 במערך הוא הממוצע של הנבדק ה-0 לאורך כל הניסוי.
4. (2 נק') - הריצו את הפונקציה מסעיף 3 על המטריצה מסעיף 2.
5. (3 נק') - השתמשו בפקודה `np.random.exponential` עם הפרמטר `size=16` וצרו מערך בשם `patients_avg_time`. הדפיסו את המערך ובדקו שאכן התקבל מערך בגודל 16.

55 נק') חלק ב' - תחקור נתונים

1. (11 נק') - כתבו את הפונקציה `max_sugar_day` אשר מקבלת כקלט מטריצה ומחזירה את היום בו ממוצע רמת הסוכר הייתה הגבוהה ביותר (בקרב כלל הנבדקים).
2. (11 נק') - כתבו את הפונקציה `max_sugar_patient` אשר מקבלת כקלט מטריצה ומחזירה את הנבדק עם ממוצע רמת הסוכר הגבוהה ביותר (בקרב כלל הנבדקים).
3. (11 נק') - כתבו את הפונקציה `get_max_time_patient` אשר מקבלת מערך חד מימדי (`patients_avg_time`) ומחזירה את מספר הנבדק אשר הזמן הממוצע עבורו הוא מקסימלי. למשל, אם הערך המקסימלי נמצא באינדקס 3 אז הפונקציה תחזיר 4 (ספירת הנבדקים מתחילה מ-0).
4. (11 נק') - כתבו את הפונקציה `count_item` המקבלת מטריצה ומספר שלם בטווח בין 0 ל `sugar_range`. הפונקציה מחזירה את כמות הפעמים שהמספר מופיע במטריצה.
5. (11 נק') - **הגדרה:** מערך מצטבר הוא מערך אשר צובר את ערכיו לאורך האינדקסים. למשל, עבור המערך `[1,2,3,4]`, המערך המצטבר הוא `[1,3,6,10]`.
- כתבו את הפונקציה `cumulative_sugar` אשר מקבלת מטריצה ומחזירה את המטריצה המצטברת. ניתן להשתמש בפונקציות מובנות של `numpy`.
6. הריצו את פונקציות 1-5 עבור אוסף הנתונים שיצרתם בחלק א'.

(30 נק') חלק ג' - ויזואליזציה

1. (15 נק') - בהתבסס על הנתונים והפונקציות מחלק ב', בחרו 3 סעיפים (פונקציות) והציגו את התוצאות בגרף בעזרת ספריית matplotlib.
יש לשמור על גרפים אינפורמטיביים: כותרת, labels, צבעים במידת הצורך וכו'.
2. (15 נק') - בצעו את הסעיפים הבאים (לפי הסדר) ליצירת גרף pie:
 - a. צרו tuple בשם labels המכיל 6 מחרוזות שונות לבחירתכם.
 - b. צרו רשימה בשם sizes, המכילה 6 מספרים אשר סוכמים ל-100, למשל [10,20,30,30,10,10].
 - c. צרו tuple בשם explode, המכיל 6 מספרים כאשר 5 מתוכם הם 0 והנוסף הוא 0.1.
(אין חשיבות לסדר האיברים).
 - d. חפשו את הפונקציה plt.pie בספריית matplotlib, השתמשו בסעיפים הקודמים והציגו גרף pie.
 - e. הסבירו במשפט מה מטרת הרשימות sizes, explode.
 - f. השתמשו בגרף pie שיצרתם בחלק זה והציגו תוצאה (איזו שתרצו) של פונקציה מחלק ב'.