

## תרגיל בית 10 (ואחרון!)

### pandas

#### הנחיות כלליות:

- קראו היטב את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- **אין לשנות את שמות הפונקציות והמשתנים שכבר מופיעים בקובץ השלד של התרגיל.**
- **אין למחוק את ההערות שמופיעות בשלד.**
- **יש לבצע בדיקה עצמית אך ורק בסוף קובץ השלד בתוך התנאי הבא:**  
`if __name__ == "__main__":`  
**בפרט אין להשאיר הדפסות/קריאות לפונקציות בין הפונקציות.**
- **אין להשאיר בקוד נתיבים לקבצים מקומיים.**
- **אין להשתמש בספריות חיצוניות (ובפונקציות שלהן) מעבר למה שסופק בשלד התרגיל.** כלומר, אין להשתמש בפקודת `import`. **כל פונקציה שלא דורשת פקודה זו מותרת לשימוש** (כלומר, זו פונקציה שהמתרגם (interpreter) מכיר ללא פקודה זו).
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ `ex10_012345678.py` המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת.
- אופן ביצוע התרגיל: בתרגיל זה עליכם לממש את הפונקציות הנתונות. ניתן להוסיף פונקציות עזר.
- מועד אחרון להגשה: כמפורסם באתר.
- היות ובדיקת התרגילים עשויה להיות אוטומטית, יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה, הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון וכי התוכנית אינה קורסת).
- בכל השאלות ניתן להניח את תקינות הקלט על פי המפורט בשאלה.

## שאלה 1 (ויחידה 😊)

”ריק (Rick) מסתובב בכל רחבי היקומים (Universes) במצוד אחר מפלצות המטילות אימה על תושביהן. הוא בוחר את יעדו לפי חישוב קר של עלות מול תועלת.

במהלך השאלה, נעזר ב2 קבצים המכילים את כלל היעדים והמפלצות אותם יש לחסל. הקבצים בפורמט CSV ומכילים את המידע הבא:

קובץ ראשון:

- בשורה הראשונה מופיעים שמות עמודות המידע בסדר הבא:
  - Universe - העמודה הראשונה; מכילה את שם היקום במצוקה.
  - Expenses - העמודה השלישית; מכילה את עלות המסע לאותו היקום ברכב חלל.
  - Duration - העמודה הרביעית; מכילה את מספר הימים שייקח להשלים את המסע (שימו לב שיכול להיות מספר לא שלם).
  - Teleporting - העמודה החמישית; מכילה אינדיקציה (ערך בוליאני) האם ניתן לעשות teleporting לאותו היקום (כלומר, להיות משוגר לשם באופן מהיר ללא צורך ברכב חלל).
- שאר השורות מכילות את המידע הרלוונטי בהתאם לעמודות כמצוין לעיל (ראו את טבלת הדוגמא למטה)
- לנוחיותכם הטבלה הבאה נמצאת כקובץ בשם "travels.csv" בפורמט csv בין קבצי התרגיל שקיבלתם:

Universe	Expenses	Duration	Teleporting
Gaia	250	2.5	False
Flip Flop	500	3	True
Tiny	100	7	True
Buya	600	3.5	False

קובץ שני:

- בשורה הראשונה מופיעים שמות עמודות המידע בסדר הבא:
  - Monster - העמודה הראשונה; מכילה את שם המפלצת שיש לחסל.
  - Bounty - העמודה השנייה; מכילה את הגמול על חיסול המפלצת.
  - Universe - העמודה השלישית; מכילה את שם היקום במצוקה.
- שאר השורות מכילות את המידע הרלוונטי בהתאם לעמודות כמצוין לעיל (ראו את טבלת הדוגמא למטה)
- לנוחיותכם הטבלה הבאה נמצאת כקובץ בשם "missions.csv" בפורמט csv בין קבצי התרגיל שקיבלתם:

Monster	Universe	Bounty
Creepy	Gaia	1000
Breepy	Flip Flop	150

Bripy	Flip Flop	150
Burpy	Tiny	500
Glupglap	Buya	250
Grumpy	Buya	500

א. ממשו את הפונקציה:

`read_files(travels_file,missions_file)`

- הפונקציה מקבלת את שמם של קבצי המידע (מטיפוס מחרוזת).
- הפונקציה תחזיר 2 אובייקטי dataframe (DF) של החבילה **pandas** לפי בסדר הבא:
  - אובייקט ראשון: טבלת המסעות בעלת 4 עמודות: Universe,Expenses,Duration,Teleporting
  - אובייקט שני: טבלת המשימות; בעלת 3 עמודות: Monster,Bounty,Universe
- במקרה של **שגיאת IO**, יש לתפוס את השגיאה ולהעלות שגיאת IO עם הכיתוב "An IO error occurred" (ראו דוגמת הרצה בעמוד הבא).
- ניתן להניח כי **במידה** ושמות הקבצים קיימים, ערכיהם יהיו תקינים ומכילים לפחות שורת משימות אחת (ושורת כותרות לעמודות).
- רמז: ניתן (אך לא חייבים) להיעזר בפקודה `pd.read_csv` על מנת לטעון את הקובץ לטבלה של **pandas**. הסתכלו בתיעוד הפונקציה באינטרנט בכדי להעביר את הארגומנטים המתאימים להשלמת השאלה.

פלט קבצי הדוגמא:

=== A ===

Travels:

```

Universe Expenses Duration Teleporting
0    Gaia      250      2.5     False
1 Flip Flop    500      3.0      True
2    Tiny     150      7.0      True
3    Buya      60      0.5     False
4    Zool     250      5.0      True

```

Missions:

```

Monster Bounty Universe
0  Creepy   280    Gaia
1  Breepy   150 Flip Flop
2   Bripy   150 Flip Flop
3   Burpy   500    Tiny
4 Glupglap   25    Buya
5  Grumpy    50    Buya
6  Grumpi   300    Zool

```

דוגמא להעלאת חריג בעקבות קלט לא תקין בסעיף א (קובץ שלא קיים)

```

FileNotFoundError: [Errno 2] No such file or directory: 'notAFile.csv'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    read_missions_file("notAFile.csv")
  File "C:\Users\LENOVO\Desktop\python2021\ex10\EX_10_sol.py", line 69, in read_missions_file
    raise IOError("An IO error occurred")
OSError: An IO error occurred

```

## מסעיף ב' ואילך ניתן להניח כי הקלט תקין

ב.

1. ממשו את הפונקציה:

`merge_tables(df_travels, df_missions)`

- הפונקציה מקבלת את פלט סעיף א' - שני אובייקטי DF (הראשון של המסעות והשני של המשימות)
- הפונקציה תמזג את 2 האובייקטים לפי עמודת שמות היקומים, ותחזיר את האובייקט הממוזג (מטיפוס DF).
- שימו לב כי בטבלה travels כל יקום מופיע פעם אחת בלבד. לעומת זאת, בטבלה missions יקום יכול להופיע מספר פעמים.
- הניחו כי לא קיים יקום המופיע בטבלה אחת בלבד.
- רמז: העזרו בפקודה `merge`
- **אין להשתמש בלולאות.**
- **יש לממש את הפונקציה בשורה אחת בלבד.**

פלט קבצי הדוגמא:

```

=== B ===
Universe Expenses Duration Teleporting Monster Bounty
0 Gaia 250 2.5 False Creepy 280
1 Flip Flop 500 3.0 True Breepy 150
2 Flip Flop 500 3.0 True Bripy 150
3 Tiny 150 7.0 True Burpy 500
4 Buya 60 0.5 False Glupglap 25
5 Buya 60 0.5 False Grumpy 50
6 Zool 250 5.0 True Grumpi 300

```

2. ממשו את הפונקציה:

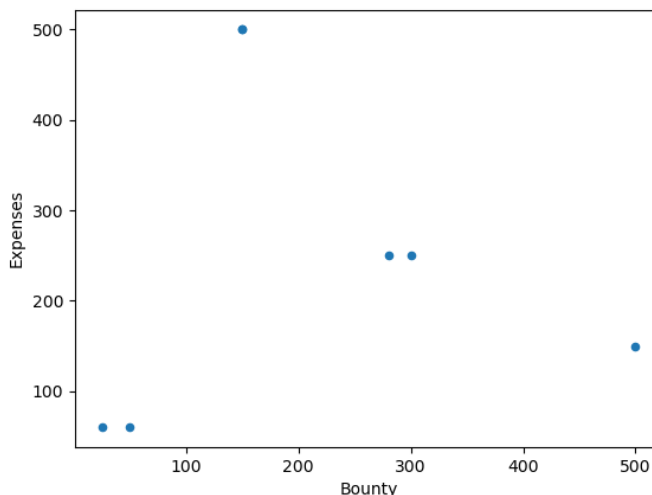
`scatter_plot(df_merged)`

- בסעיף זה נצייר דיאגרמה מסוג `scatter plot`

○ לקריאה כללית ראו: [https://en.wikipedia.org/wiki/Scatter\\_plot](https://en.wikipedia.org/wiki/Scatter_plot)

○ למימוש scatter plot באמצעות pandas ראו: <https://pandas.pydata.org/pandas-docs/version/0.25.0/reference/api/pandas.DataFrame.plot.scatter.html>

- הפונקציה מקבלת את פלט סעיף ב.1.
- הפונקציה תצייר scatter plot, **ותחזיר את האג** (אובייקט מסוג SubplotBase) עליו התבצע הצירוף.
- להלן תוצאת הצירוף המבוקשת:



ג. ממשו את הפונקציה:

**add\_gain\_per\_monster(df\_merged)**

- קלט הפונקציה הינו טבלת pandas הפלט מסעיף ב'.
- על הפונקציה להוסיף **לטבלה הקיימת** עמודה בשם Gain המכילה את הרווח מחיסול כל מפלצת. הרווח נמדד לפי תגמול (Bounty) בניקוי ההוצאות נסיעה (Expenses).
- **לדוגמא:** בהינתן מערך כפי שמופיעה בטבלה לעיל, כדאיות המשימה ביקום Gaia היא 750 מכיוון ש- $280 - 250 = 30$
- העמודה החדשה תיקרא "Gain"
- בשורות בהם לא ניתן להסיק את ערך gain עקב ערכים חסרים (Bounty או Expenses) יש לשים **בעמודה Gain בלבד** את הערך 0.
- **אין להשתמש בלולאות.**
- **יש לממש את הפונקציה בשורה אחת בלבד.**
- **הפונקציה לא צריכה להחזיר כלום.**

ערך הטבלה לאחר העדכון (לפי קבצי הדוגמא):

=== C ===

	Universe	Expenses	Duration	Teleporting	Monster	Bounty	Gain
0	Gaia	250	2.5	False	Creepy	280	30
1	Flip Flop	500	3.0	True	Breepy	150	-350
2	Flip Flop	500	3.0	True	Bripy	150	-350
3	Tiny	150	7.0	True	Burpy	500	350

4	Buya	60	0.5	False	Glupglap	25	-35
5	Buya	60	0.5	False	Grumpy	50	-10
6	Zool	250	5.0	True	Grumpi	300	50

על מנת ליעל את פעולותיו, החליט ריק כי לכל יקום בו ייסע, יהרוג את כל המפלצות באותה הנסיעה, וכך ישקיע זמן נסיעה (Duration) ישלם הוצאות נסיעה (Expenses) פעם אחת בלבד עבור חיסול כל המפלצות מאותו היקום.

ד. ממשו את הפונקציה:

`daily_gain_per_universe(df_merged)`

- קלט לפונקציה הינו הטבלה שהוחזרה בסעיף ב'.
- על הפונקציה להחזיר עמודה מסוג סדרה (Series).
  - כותרת פלט הסדרה היא Universe.
  - האינקסים של העמודה הם שמות היקום
  - הערכים של העמודה הם הרווח הנקי היומי עבור כל יקום
  - טיפוס הערכים בסדרה הוא float64.
- הרווח היומי הנקי עבור כל יקום מחושב סך הרווחים מחיסול כל מפלצות באותו היקום (כפי שחישבתם בסעיף ג') חלקי סך הימים אשר יידרשו להשלמת כלל המשימות באותו היקום.
- לדוגמא: בהינתן הטבלאות בדוגמא לעיל, קיימת משימה אחת ביקום Gaia וכדאיותה היא 12 מכיוון ש- $\frac{280-250}{2.5} = 12$ .
- דוגמא נוספת: בהינתן הטבלאות בדוגמא לעיל, קיימות 2 משימות ביקום Buya ולכן כדאיות יקום זה היא 30 מכיוון ש- $\frac{25+50-60}{0.5} = 30$ .
- רמז: השתמשו בgroupby
- אין להשתמש בלולאות.
- יש לממש את הפונקציה בשורה אחת בלבד.

פלט מקבצי הדוגמא:

```

=== D ===
Universe
Buya      30.000000
Flip Flop -66.666667
Gaia      12.000000
Tiny      50.000000
Zool      10.000000
dtype: float64

```

ה. ממשו את הפונקציה:

`drop_nonprofitable_universes(df_merged)`

- קלט לפונקציה הינו הטבלה שהוחזרה בסעיף ב'.

- על הפונקציה להחזיר טבלה חדשה המכילה את כל היקומים עבורם סך הרווח יומי חיובי (כפי חושב בסעיף הקודם).
- לדוגמא, עבור ביקום Flip Flop קיימות 2 מפלצות, אך סך התגמול שלהן (300) קטן מהוצאות נסיעה ליקום זה (500), ולכן השורות המכילות את היקום Flip Flop יוסרו מהטבלה.
- במידה וקיימת יותר ממפלצת אחת באותו היקום, הוצאות המסע יחושבו פעם אחת בלבד.
- ניתן (ורצוי) להשתמש בפונקציות שמימשותם בסעיפים הקודמים.

===E===

	Universe	Expenses	Duration	Teleporting	Monster	Bounty
0	Gaia	250	2.5	False	Creepy	280
3	Tiny	150	7.0	True	Burpy	500
4	Buya	60	0.5	False	Glupglap	25
5	Buya	60	0.5	False	Grumpy	50
6	Zool	250	5.0	True	Grumpi	300

1. ממשו את הפונקציה:

`mean_duration(df_5)`

- קלט לפונקציה הינו הטבלה שהוחזרה בסעיף ה'.
- את ממוצע זמן הנסיעות הנדרש עבור חיסול כלל המפלצות
- תזכורת: במידה וקיימת יותר ממפלצת אחת באותו היקום, הוצאות המסע יחושבו פעם אחת בלבד.
- אין להשתמש בלולאות
- יש לכתוב את גוף הפונקציה בשורה אחת.

===F===

3.75

2. ממשו את הפונקציה:

`drop_least_daily_lucrative_universe(df_5)`

- קלט לפונקציה הינו סדרת הפלט של מסעיף ה'.
- על הפונקציה להחזיר 3 פלטים (מופרדים באמצעות פסיק) בסדר הבא:
  - את שם היקום (מסוג מחרוזת) בו הרווח היומי (שחושב סעיף הקודם) הכי פחות כדאי (הכי נמוך).
  - ערך הרווח היומי הכי פחות כדאי (מטיפוס float).
  - עותק של טבלת הקלט ללא השורה של היקום בעל הרווח הנמוך ביותר.
- ניתן להניח כי קיים יקום אחד בעל רווח יומי מינימלי.

===G===

	Universe	Expenses	Duration	Teleporting	Monster	Bounty
0	Gaia	250	2.5	False	Creepy	280
3	Tiny	150	7.0	True	Burpy	500
4	Buya	60	0.5	False	Glupglap	25
5	Buya	60	0.5	False	Grumpy	50

ח. ריק השיג טלפורטר (portal gun), וכעת הוא יכול לשגר את עצמו באמצעות teleporting לכל יקום אשר המכיל True בעמודה teleporting.

השימוש בteleporting מקצר את משך המסע (duration) בחצי. לדוגמא, מסע לTiny יתקצר מ7 ל3.5.

ממשו את הפונקציה `drop_least_daily_lucrative_universe_with_teleporting(df_merged)`

קלט הפונקציה הוא טבלת הפלט מסעיף ב'.

על הפונקציה לפתור שוב את סעיף ז' תוך הנחה שריק השתמש ב portal gun לכל יקום אליו יכול לעשות זאת.

הנחיות:

- תחילה, הפונקציה תעדכן את העמודה Duration לפי עבור כל שורה בה הערך Teleporting הוא True ולאחר מכן תבצע את שאר החישוב.
- ניתן (ואף רצוי) להשתמש בסעיפים הקודמים לפתרון סעיף זה.

===H===

	('Gaia', 12.0,	Universe	Expenses	Duration	Teleporting	Monster	Bounty
3	Tiny	150	3.5	True	Burpy	500	
4	Buya	60	0.5	False	Glupglap	25	
5	Buya	60	0.5	False	Grumpy	50	
6	Zool	250	2.5	True	Grumpi	300	)