

## תרגיל כתיבה: עבודה עם DataFrame

בתרגיל זה תתנסו בבניית טבלה מתוך קובץ המכיל תוצאות משחקי כדורגל ובהוצאת נתונים כלליים. הורידו מה-Moodle את הקובץ 2014.results.txt שמכיל את כל תוצאות ה-Premier League (הליגה האנגלית העליונה) בעונת 2014. כל קבוצה משחקת נגד כל קבוצה אחרת פעמיים בעונה – פעם בתור הקבוצה הביתית ופעם בתור הקבוצה האורחת. מבנה הקובץ הוא:

1. שם הקבוצה הביתית (Home)
2. שם הקבוצה האורחת (Away)
3. מספר השערים שהובקעו ע"י הקבוצה הביתית (Goals\_home)
4. מספר השערים שהובקעו ע"י הקבוצה האורחת (Goals\_away)

קבוצה מנצחת אם היא מבקיעה יותר שערים מהקבוצה השנייה, מסיימת בתיקו אם שתי הקבוצות הבקיעו מספר שערים זהה, ומפסידה אם הקבוצה השנייה הבקיעה יותר שערים. למשל: בשורה הבאה מתוך הקובץ:

```
QPR Hull 0 1
```

הקבוצה הביתית היא QPR, הקבוצה האורחת היא Hull, Hull ניצחה את QPR 1-0.

בצעו את הפעולות הבאות:

1. קראו את קובץ התוצאות לטבלה בעזרת `read_csv` של `pandas`. הקובץ הוא `tab-delimited` והשורה הראשונה היא שורת `header`, השתמשו בפרמטרים המתאימים כדי לציין את זה ב-`read_csv`.

### חלק א: נתונים כלליים על הליגה

בחלק זה אין להשתמש בלולאות כלל! השתמשו בפונקציות של `DataFrame` ככל הניתן.

2. הדפיסו את מספר השערים הגבוה ביותר שהובקע על ידי קבוצה ביתית. ציינו מי הקבוצה ונגד מי היא שיחקה באותו משחק. השתמשו במתודה `argmax` של `Series` שמחזירה את האינדקס של הערך הגבוה ביותר.
3. הדפיסו את מספר השערים הגבוה ביותר שהובקע במשחק כלשהו. ציינו מי הן הקבוצות ומה היתה התוצאה.
4. הדפיסו את מספר השערים הממוצע שמבקיעות קבוצות ביתיות ואת מספר השערים הממוצע שמבקיעות קבוצות אורחות.
5. הדפיסו את מספר המשחקים שבהם ניצחה הקבוצה הביתית, האורחת וכן מספר המשחקים שהסתיימו בתיקו.
6. הדפיסו את מספר השערים שכל קבוצה הבקיעה לאורך כל העונה. לצורך ביצוע משימה זו השתמשו במתודה `groupby` של `DataFrame`. המתודה `groupby` מקבלת שם של עמודה אחת או יותר ומקבצת את השורות בטבלה לפי העמודה. למשל: אם תעבירו ל-`groupby` את `Home` אז המתודה תחזיר טבלה שבה השורות מחולקות ל-20 קבוצות (סה"כ יש 20 קבוצות בליגה האנגלית). מתוך ה"טבלה המקובצת" אפשר לבחור עמודה (או יותר) ועליה לבצע פעולה שתתבצע על כל קבוצה בנפרד. למשל:

```
print(results.groupby('Home')['Goals_home'].mean())
```

מחזירה Series שבו ה-index מורכב משמות הקבוצות והערכים הם ממוצעי ההבקעות במשחקי הבית:

```
Home
Arsenal      2.157895
Aston_Villa  0.947368
Burnley      0.736842
Chelsea      1.894737
Crystal_Palace 1.105263
Everton      1.421053
Hull         1.000000
Leicester    1.473684
Liverpool    1.578947
Man_City     2.315789
Man_United   2.157895
Newcastle    1.368421
QPR          1.210526
Southampton  1.947368
Stoke        1.684211
Sunderland   0.842105
Swansea      1.421053
Tottenham    1.631579
West_Brom    1.263158
West_Ham     1.315789
Name: Goals_home, dtype: float64
```

7. הדפיסו את סך כל השערים שהובקעו במשחקי הבית של כל קבוצה. לצורך ביצוע משימה זו השתמשו במתודה groupby של DataFrame ועל ה"טבלה המקובצת" שאותה היא מחזירה הפעילו את המתודה apply עם lambda function. apply מקבלת פונקציה ומפעילה אותה על כל קבוצה בנפרד. פונקציית lambda מתאימה למצבים שבהם אין פונקציה קיימת שעושה את העבודה. הפונקציה מקבלת את החלק בטבלה שמתייחס לכל קבוצה ומבצעת עליו את הקוד הנדרש. למשל: הקוד הבא מבצע את אותה פעולה כמו בדוגמא של הסעיף הקודם:

```
print(results.groupby('Home').apply(lambda x: x['Goals_home'].mean()))
```

### חלק ב': בניית טבלה

השתמשו בפעולות השונות של DataFrame בשביל לבנות DataFrame חדש שיכיל את טבלת הליגה האנגלית בסיום עונת 2014. האינדקס של הטבלה יהיה שמות הקבוצות, הטבלה צריכה לכלול את העמודות הבאות:

מספר ניצחונות (Wins), מספר תוצאות תיקו (Ties), מספר הפסדים (Losses), סך כל השערים שהקבוצה הבקיעה (GF), סך כל השערים שהקבוצה ספגה (שהבקיעו נגדה, GA), מספר נקודות (Points).

קבוצות מקבלות 3 נקודות על ניצחון, נקודה אחת על תיקו ואפס נקודות על הפסדים. הטבלה צריכה להיות ממוינת לפי מספר נקודות. קבוצות שסיימו עם אותו מספר נקודות יכולות להופיע בכל סדר שהוא. הגישו את הקובץ עם הקוד (או jupyter notebook) לא מכוון דרך המודל.