



מבוא לתכנות מדעי וסטטיסטי – R

חוברת תרגילי כיתה

1. משתנים, השמות, וקטורים ומטריצות

1. נתונים שני משתנים מספריים, x ו- y ; כתבו סדרת פקודות שבסופה התוכן של שני המשתנים יתחלף.

2. המשתנה x מכיל את ציון בוחן האמצע של סטודנט בקורס, והמשתנה y מכיל את ציון בחינת מועד א'. הציון הסופי בקורס מורכב מ-25% בוחן אמצע, ו-75% מועד א'. כתבו פקודה שתוצאתה TRUE אם הסטודנט עבר את הקורס (הציון הסופי שלו הוא 51 או יותר), ו-FALSE אחרת.

3. כתבו פקודה שתוצאתה היא הווקטור הבא:

[1] 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

4. נתונים שני וקטורים מספריים, x ו- y (לא בהכרח באותו האורך). כתבו פקודה שתוצאה היא TRUE אם כל האיברים שב- x גדולים מכל האיברים שב- y , ו-FALSE אחרת.

5. נתון כי x הוא וקטור. כתבו פקודה שתוצאתה היא האיבר האחרון בווקטור.

6. נתון כי x הוא וקטור. כתבו פקודה שתוצאתה היא וקטור שמכיל את כל האיברים עם האינדקס הזוגי שב- x (כלומר האיבר השני, הרביעי, השישי, וכו').

7. נתון כי x הוא וקטור. כתבו פקודה שתוצאתה היא וקטור עם כל איברי x שאינם NaN.

8. נתון כי x הוא וקטור מספרי. כתבו פקודה המוצאת כמה איברים ב- x מתחלקים ב-3 ללא שארית.

9. הגדרה: הממוצע הגיאומטרי של המספרים x_1, x_2, \dots, x_n הוא $\sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$. כתבו פקודה המחשבת את הממוצע הגיאומטרי של וקטור מספרי x .

10. הגדרה: הממוצע ההרמוני של המספרים x_1, x_2, \dots, x_n הוא

$$\frac{n}{1/x_1 + 1/x_2 + \dots + 1/x_n}$$

כתבו פקודה המחשבת את הממוצע ההרמוני של וקטור מספרי x .

11. נתון כי x הוא וקטור מספרי. כתבו פקודה המחליפה את האיבר המינימלי שב- x במספר 17.

12. כתבו פקודה היוצרת את המטריצה הבאה:

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]	[, 6]	[, 7]	[, 8]	[, 9]	[, 10]
[1,]	1	1	1	1	1	1	1	1	1	1
[2,]	2	2	2	2	2	2	2	2	2	2
[3,]	3	3	3	3	3	3	3	3	3	3
[4,]	4	4	4	4	4	4	4	4	4	4
[5,]	5	5	5	5	5	5	5	5	5	5

13. נתון כי A היא מטריצה. כתבו פקודה שתוצאתה היא TRUE אם ל- A יש יותר שורות מאשר עמודות, ו-FALSE אחרת.

14. נתון כי A היא מטריצה מספרית. כתבו פקודה שתוצאתה היא TRUE אם כל אברי A הם חיוביים, ו-FALSE אחרת.

15. נתון כי A היא מטריצה בת שתי שורות לפחות. כתבו פקודה שתוצאתה היא השורה הלפני אחרונה של A.

16. נתון כי A היא מטריצה. כתבו פקודה שתוצאתה היא A בסדר שורות הפוך (כלומר השורה האחרונה הופכת לראשונה, וכו').

17. נתון כי A היא מטריצה מספרית. כתבו פקודה שתוצאתה היא הסכום של המספרים המקסימליים בכל עמודה ב-A (כלומר המקסימום בעמודה הראשונה, ועוד המקסימום בשנייה, וכו').

18. נתון כי A היא מטריצה מספרית. כתבו פקודה שתוצאתה היא האינדקס של העמודה שבה נמצא המספר המקסימלי במטריצה.

19. כתבו פקודה שתוצאתה היא מטריצה אלכסונית בת 5 שורות ו-5 עמודות, שבאלכסון שלה מופיע 5 פעמים המספר 17 :

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]
[1,]	17	0	0	0	0
[2,]	0	17	0	0	0
[3,]	0	0	17	0	0
[4,]	0	0	0	17	0
[5,]	0	0	0	0	17

20. כתבו פקודה שתוצאתה היא מטריצה בת 5 שורות ו-5 עמודות שכולה אפסים, ורק ב"מסגרת" שלה יש אחדות :

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]
[1,]	1	1	1	1	1
[2,]	1	0	0	0	1
[3,]	1	0	0	0	1
[4,]	1	0	0	0	1
[5,]	1	1	1	1	1

2. פונקציות, תנאים, לולאות ורקורסיה

1. כתבו פונקציה המקבלת כארגומנט מטריצה מספרית, ומחזירה את האינדקס של העמודה במטריצה שסכום האיברים שבה מקסימלי.

2. כתבו פונקציה המקבלת כארגומנט וקטור מספרי, ומחזירה T אם הוא וקטור קבוע (כלומר כל איבריו שווים זה לזה), ו-F אחרת.

3. כתבו פונקציה המקבלת כארגומנטים שני וקטורים, ומחזירה T אם הם שווים, ו-F אחרת.

4. כתבו פונקציה בשם my.max המקבלת כארגומנט וקטור מספרי ומחזירה את האיבר המקסימלי שבו (כלומר עושה את מה ש-max עושה עבור וקטורים), בלי להשתמש בפונקציה max.

5. כתבו פונקציה בשם sec.diag המקבלת כארגומנט מספר טבעי n, ומחזירה מטריצה ריבועית מסדר n שכל אבריה אפסים, פרט לאלכסון המשני (מהפינה השמאלית התחתונה לימנית העליונה), שכל אבריו הם 1.

```
> sec.diag(3)
      [,1] [,2] [,3]
[1,]    0    0    1
[2,]    0    1    0
[3,]    1    0    0
```

6. כתבו פונקציה **רקורסיבית** בשם `recursive.sum` המקבלת כארגומנט וקטור מספרי, ומחזירה את סכום איברי הווקטור (כלומר עושה את מה ש-`sum` עושה עבור וקטורים).

7. הגדרה: **סדרת פיבונאצ'י** (Fibonacci) היא סדרה ששני אבריה הראשונים הם 1, וכל אחד מהאיברים הבאים הוא הסכום של השניים שקדמו לו:

$$x_1 = 1, \quad x_2 = 1, \quad x_n = x_{n-1} + x_{n-2}, \quad n > 2$$

תחילת הסדרה היא 1,1,2,3,5,8,13,...

כתבו פונקציה בשם `fibonacci` המקבלת כארגומנט מספר טבעי n , ומחזירה את האיבר ה- n בסדרת פיבונאצ'י.

8. כתבו פונקציה בשם `f` המקבלת כארגומנט מטריצה מספרית, ומחזירה את המטריצה כשכל האיברים ה"פנימיים" שלה הופכים ל-0. למשל, אם `A` היא המטריצה הבאה,

```
> A
      [,1] [,2] [,3] [,4] [,5]
[1,]     4     9     2     2     1
[2,]     7     1     7     7     8
[3,]     2     1     5     5     5
[4,]     3     6     3     4     6
```

או

```
> f(A)
      [,1] [,2] [,3] [,4] [,5]
[1,]     4     9     2     2     1
[2,]     7     0     0     0     8
[3,]     2     0     0     0     5
[4,]     3     6     3     4     6
```

ניתן להניח שבמטריצת הקלט יש לפחות 3 שורות ולפחות 3 עמודות.

3. מחרוזות, קבצים, קלט ופלט

1. כתבו פונקציה בשם `count.char`, המקבלת כארגומנט ראשון מחרוזת וכארגומנט שני מחרוזת של תו **בודד**, ומחזירה את מספר הפעמים שהתו מופיע במחרוזת. למשל,

```
> count.char("abracadabra", "b")
[1] 2
```

2. כתבו פונקציה בשם `f` המקבלת כארגומנט מחרוזת, ומחזירה את מספר התווים במחרוזת **שאינם** אותיות (גדולות או קטנות). למשל,

```
> f("$a%B")
[1] 3
> f("Ab C")
[1] 1
```

3. כתבו פונקציה בשם `draw.vector` המקבלת כארגומנט וקטור של מספרים טבעיים, ומדפיסה לקונסולה שורות של כוכביות, שאורכיהן הם המספרים שבוקטור. למשל,

```
> draw.vector(c(3, 5, 1, 2))
***
*****
*
**
```

4. כתבו פונקציה בשם `print.sum` המקבלת כארגומנט וקטור של מספרים, ומדפיסה את סכום האיברים שלו באופן הבא:

```
> print.sum(c(4, 7, 2))
4 + 7 + 2 = 13
```

5. כתבו פונקציה בשם `star.files` המקבלת כארגומנט מספר טבעי n , ואז יוצרת n קבצים: לקובץ הראשון קוראים `file1.txt`, והוא מכיל כוכבית אחת; לשני קוראים `file2.txt`, והוא מכיל שתי כוכביות (בשורה); וכו'.

6. כתבו פונקציה בשם `f` המקבלת כארגומנט ראשון מחרוזת `s` וכארגומנט שני תו בודד `ch`, ומחזירה את המיקום של `ch` במחרוזת `s` (כלומר אם `ch` הוא התו הראשון ב-`s` הפונקציה תחזיר 1, אם הוא שני תחזיר 2, וכו'). אם `ch` מופיע יותר מפעם אחת ב-`s`, הפונקציה צריכה להחזיר את המיקום של המופע הראשון. אם `ch` לא מופיע בכלל, הפונקציה צריכה להחזיר 0.

למשל,

```
> f("university", "i")
[1] 3
> f("university", "w")
[1] 0
```

4. דוגמאות תכנות

1. "ראשוניים תאומים" הם זוג מספרים ראשוניים שההפרש ביניהם הוא 2, כמו הזוג 3 ו-5, או הזוג 11 ו-13. כתבו פונקציה בשם `twin.primes` המקבלת כארגומנטים שני מספרים טבעיים `a` ו-`b`, ומדפיסה למסך את כל זוגות הראשוניים התאומים שבין `a` ל-`b`, באופן הבא:

```
> twin.primes(1, 20)
3 , 5
5 , 7
11 , 13
17 , 19
> twin.primes(10, 20)
11 , 13
17 , 19
```

ניתן להשתמש בפונקציה `is.prime` שנלמדה בהרצאה, ושנמצאת באתר הקורס.

2. "מספר משוכלל" הוא מספר ששווה לסכום של המחלקים שלו (כולל 1, ולא כולל המספר עצמו). למשל, 6 הוא מספר משוכלל כי 6 מתחלק ב-1, 2, ו-3, ומתקיים השוויון $1 + 2 + 3 = 6$. כתבו פונקציה בשם `perfect.number` המקבלת כארגומנט מספר טבעי, ומחזירה TRUE אם הוא משוכלל, ו-FALSE אחרת. למשל,

```
> perfect.number(6)
[1] TRUE
> perfect.number(10)
[1] FALSE
```

3. כתבו פונקציה בשם `digit.sum` המקבלת כארגומנט מספר טבעי, ומחשבת שוב ושוב את סכום הספרות שלו, עד שמתקבל מספר חד ספרתי. למשל, אם המספר הוא 167, סכום הספרות הוא 14, וסכום הספרות של 14 הוא 5, שהוא כבר מספר חד ספרתי, ולכן התשובה היא 5.

```
> digit.sum(167)
[1] 5
```

על הפונקציה להיות **רקורסיבית**.

4. כתבו פונקציה בשם `decimal.to.roman` המקבלת כארגומנט מספר טבעי בין 1 ל-99, ומחזירה מחרוזת המתאימה למספר בספרות רומיות. למשל,

```
> decimal.to.roman(2)
[1] "II"
> decimal.to.roman(60)
[1] "LX"
> decimal.to.roman(99)
[1] "XCIX"
```

5. ספרת הביקורת בתעודת הזהות נקבעת על סמך הספרות שמשמאלה. כדי להסביר איך הדבר נעשה, ניקח כדוגמה את מספר הזהות 32578349.

שלב 1: מכפילים את ספרות מספר הזהות במספרים 1 ו-2, לסירוגין:

$$3 \times 1 \quad 2 \times 2 \quad 5 \times 1 \quad 7 \times 2 \quad 8 \times 1 \quad 3 \times 2 \quad 4 \times 1 \quad 9 \times 2 \quad \rightarrow \quad 3 \quad 4 \quad 5 \quad 14 \quad 8 \quad 6 \quad 4 \quad 18$$

הערה: בתחילת השלב, מספר הזהות צריך להיות בן 8 ספרות, ולעתים צריך להוסיף משמאלו אפס אחד או יותר על מנת שיהיה כזה.

שלב 2: מחליפים מספרים דו-ספרתיים בסכום הספרות שלהם:

$$3 \quad 4 \quad 5 \quad 14 \quad 8 \quad 6 \quad 4 \quad 18 \quad \rightarrow \quad 3 \quad 4 \quad 5 \quad 5 \quad 8 \quad 6 \quad 4 \quad 9$$

שלב 3: מחברים את המספרים שהתקבלו:

$$3 + 4 + 5 + 5 + 8 + 6 + 4 + 9 = 44$$

שלב 4: ספרת הביקורת היא ההפרש בין הסכום האחרון לבין הכפולה השלמה של 10 הקרובה ביותר **מלמעלה** לסכום הנ"ל:

$$50 - 44 = 6$$

אילו היינו מקבלים בשלב 3 את הסכום 70, למשל, אז ספרת הביקורת הייתה 0.

א. כתבו פונקציה בשם `find.check.digit` המקבלת כארגומנט מספר זהות **ללא** ספרת הביקורת, ומחזירה את ספרת הביקורת המתאימה. למשל,

```
> find.check.digit(32578349)
[1] 6
```

ב. כתבו פונקציה בשם `verify.check.digit` המקבלת כארגומנט מספר זהות **כולל** ספרת הביקורת, מחזירה `TRUE` אם ספרת הביקורת היא נכונה, ו-`FALSE` אחרת. הפונקציה יכולה כמובן לקרוא לפונקציה `find.check.digit` מסעיף א'. למשל,

```
> verify.check.digit(325783496)
[1] TRUE
```

5. רשימות, מסגרות נתונים ופקטורים

1. נניח שהמשתנה `x` הוא רשימה בת שני פריטים, בה הפריט הראשון הוא שם של אדם, והפריט השני הוא וקטור של הגילים של הילדים שלו. למשל,

```
> x <- list("Charlie", c(2, 6.5, 11))
> x
[[1]]
[1] "Charlie"

[[2]]
[1] 2.0 6.5 11.0
```

כתבו פונקציה בשם `f` המקבלת כארגומנט משתנה כדוגמת `x`, ומדפיסה את שם האדם, את מספר הילדים שלו, ואת גיל הילד הצעיר ביותר, באופן הבא:

```
> f(x)
Charlie has 3 children.
The youngest is 2 years old.
```

2. בקובץ `diets.R` נמצאים נתונים אודות ניסוי המשווה את היעילות של שתי דיאטות הרזייה זו ביחס זו לזו, וכן ביחס לקבוצת ביקורת (שהיא קבוצת נבדקים שלא השתתפה באף דיאטה). תוכן העמודות הוא

age – גיל הנבדק (בשנים)
height – גובה הנבדק (בס"מ)
sex – מין הנבדק (M = זכר; F = נקבה)
smoking – מספר סיגריות ממוצע ליום שהנבדק מעשן
weight.before – משקל (בק"ג) לפני תחילת הדיאטה
weight.after – משקל (בק"ג) לאחר תום הדיאטה
treatment – סוג הדיאטה שעבר הנבדק (0 = קבוצת ביקורת; 1 = דיאטה א'; 2 = דיאטה ב')

טענו את הקובץ למסגרת נתונים בשם `diets`, וענו על השאלות הבאות.

א. הפכו את העמודה `treatment`, שהיא כעת עמודה מספרית, לפקטור בן שלוש רמות ששמותיהן `A`, `control` ו-`B`. הציגו את סיכום מסגרת הנתונים המעודכנת באמצעות הפונקציה `summary`.

ב. מהו ממוצע גיל הנשים, ומהו ממוצע גיל הגברים?

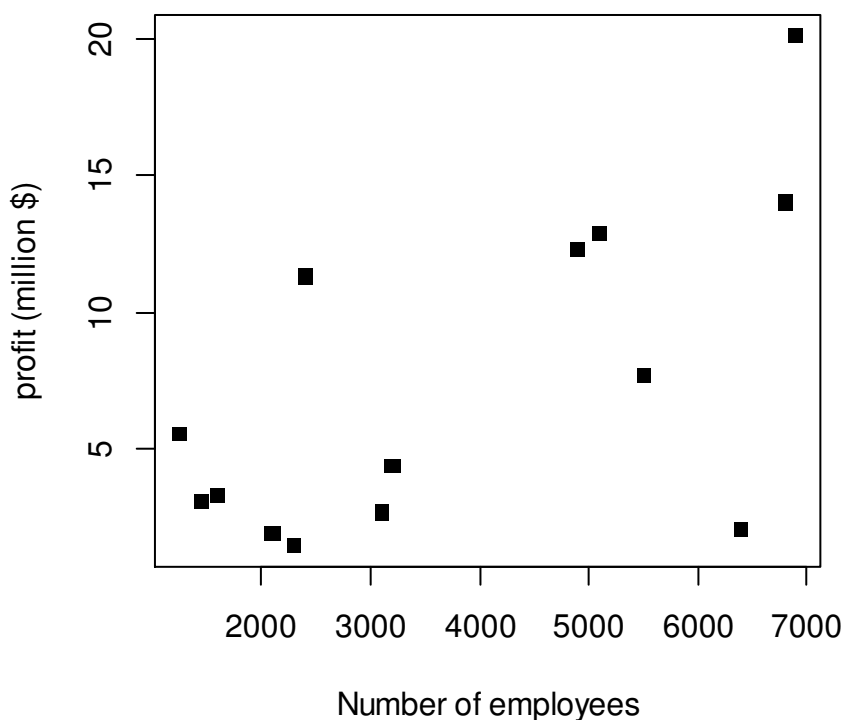
ג. מהו **שינוי** המשקל (משקל סופי פחות משקל התחלתי) הממוצע במהלך הניסוי בקרב כל אחת משלוש קבוצות הטיפול (קבוצת ביקורת, דיאטה א' ודיאטה ב')?

ד. אינדקס מסת הגוף של אדם (בקיזור – BMI) הוא המשקל (בק"ג) שלו חלקי ריבוע הגובה (במטרים, ולא בסנטימטרים). למשל, נבדק שגובהו 187 ס"מ ומשקלו 76 ק"ג הוא בעל אינדקס מסת גוף של $76/(1.87)^2 = 21.73$. הוסיפו למסגרת הנתונים שתי עמודות חדשות בשם BMI.before ו-BMI.after, המתארות את אינדקס מסת הגוף ההתחלתי והסופי של הנבדקים.

ה. הוסיפו למסגרת הנתונים עמודה חדשה, מטיפוס פקטור, בשם smoking.category. לפקטור יהיו ארבע רמות: none – עבור נבדקים שאינם מעשנים כלל; light – עבור נבדקים המעשנים בממוצע בין סיגריה אחת לחמש ביום; medium – עבור נבדקים המעשנים בממוצע בין שש ל-20 סיגריות ביום; ו-heavy – עבור נבדקים המעשנים בממוצע 21 סיגריות או יותר ביום.

6. גרפיקה ב-R

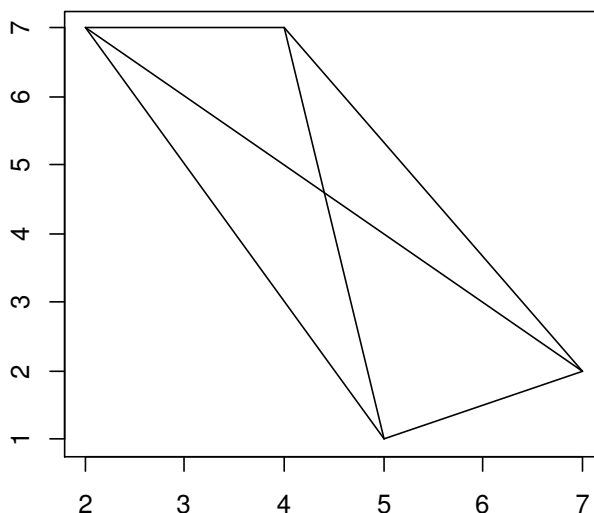
1. בקובץ profits.txt נמצאים נתונים אודות מספר העובדים והרווח השנתי (במיליוני דולרים) של 14 חברות אמריקאיות. צרו דיאגרמת פיזור המציגה את הנתונים באופן הבא:



2. כתבו פונקציה בשם `plot_all_lines` המקבלת כארגומנטים שני וקטורים מספריים באותו האורך, ומשרטטת את כל הקווים בין הקואורדינטות שהם יוצרים (כאשר הארגומנט הראשון הוא וקטור של ערכי x , והשני של ערכי y). למשל, הפקודה

```
plot_all_lines(c(4, 2, 5, 7), c(7, 7, 1, 2))
```

תיצור את הגרף

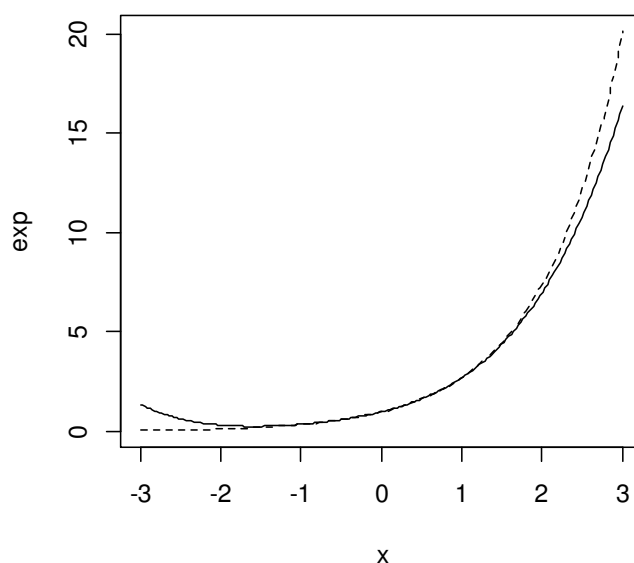


3. תזכורת: קירוב טיילור מסדר n של הפונקציה e^x (סביב הנקודה 0) הוא

$$e^x \approx 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

כתבו פונקציה בשם `taylor_approx` המקבלת כארגומנט מספר טבעי n , ויוצרת גרף מקווקו של הפונקציה e^x בתחום שבין -3 ל-3, יחד עם גרף של קירוב טיילור מסדר n של הפונקציה. כותרת הגרף צריכה לציין את הערך של n . למשל, הפקודה `taylor_approx(4)` תיצור את הגרף הבא:

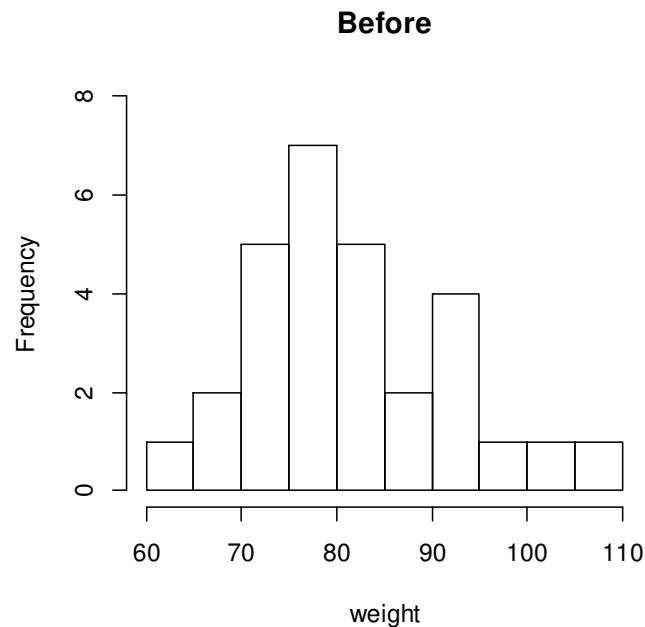
n = 4



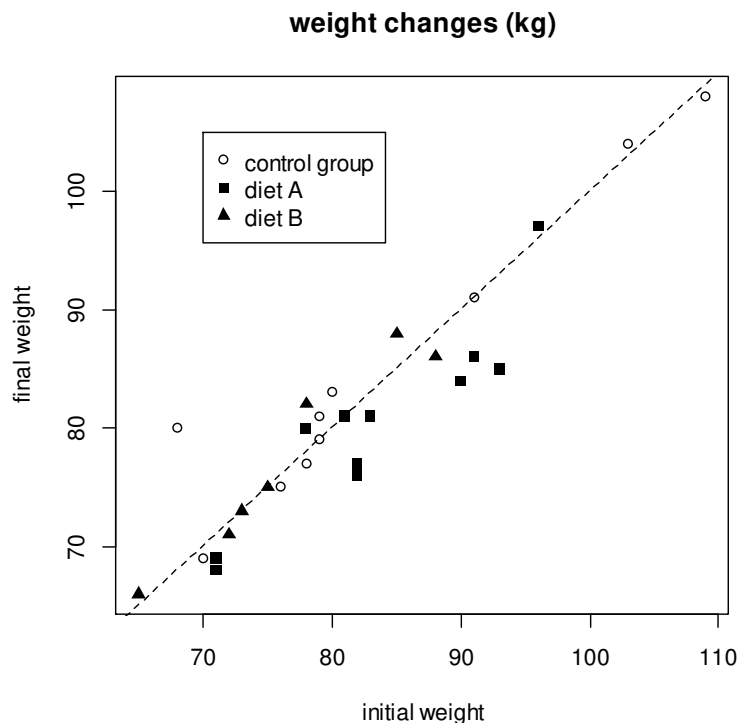
מומלץ להריץ את הפונקציה עם ערכי n בין 2 ל-7, ולראות כיצד הקירוב לפונקציה e^x הולך ומשתפר ככל ש- n גדל.

4. שאלה זו היא המשך של שאלה 2 מהתרגול הקודם, אודות הניסוי המשווה את היעילות של שתי דיאטות הרזייה (הנתונים נמצאים בקובץ diets.R).

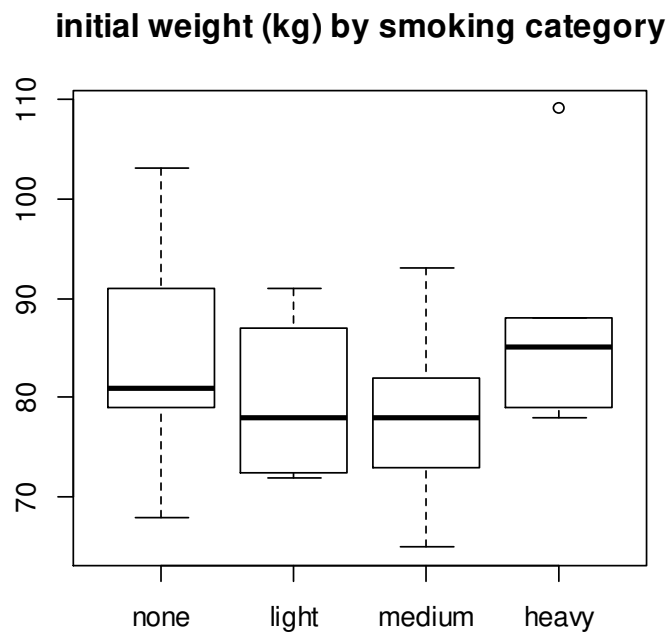
א. הכינו היסטוגרמה של התפלגות המשקל ההתחלתי בקרב כל משתתפי הניסוי, הנראית כך:



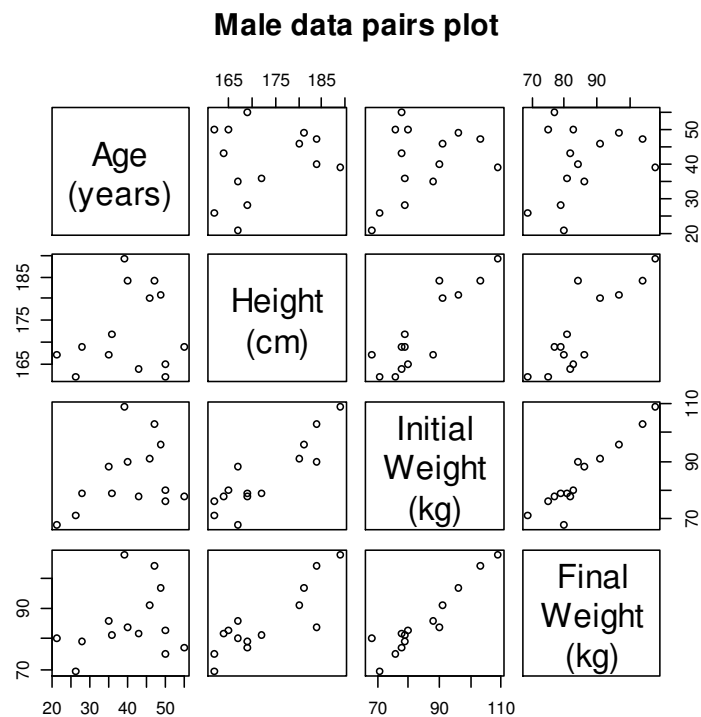
ב. הכינו דיאגרמת פיזור שהציר האופקי שלה הוא המשקל ההתחלתי, הציר האנכי שלה המשקל הסופי, וכל קבוצת טיפול מיוצגת על-ידי סמל שונה. הוסיפו לדיאגרמה קו מקווקו המתאים למשוואה $y = x$, באמצעותו קל יותר להבחין אילו נבדקים עלו במשקל ואילו ירדו. מהן המסקנות העולות מהדיאגרמה?



ג. הכינו ארבעה תרשימי קופסה המוצגים זה לצד זה, המתארים את התפלגות המשקל ההתחלתי בקרב ארבע הרמות של הפקטור `smoking.category`.



ד. הכינו תרשים המציג, עבור הנבדקים **הגברים בלבד**, דיאגרמת פיזור עבור כל זוגות העמודות האפשריים מבין `age`, `height`, `weight.before`, `weight.after`.



5. היזכרו ב"סדרות קולאץ" בהן נתקלנו בהרצאה.

א. כתבו פונקציה בשם `collatz.report` המקבלת כארגומנט מספר טבעי n , ומחזירה רשימה בת שלושה פריטים:

- הפריט הראשון, ששמו `sequence`, הוא סדרת קולאץ' שמתחילה ב- n , ומסתיימת ב-4 הראשון
- הפריט השני, ששמו `length`, הוא אורך הסדרה (כלומר מספר האיברים שבה)
- הפריט השלישי, ששמו `maximum`, הוא האיבר המקסימלי בסדרה

למשל,

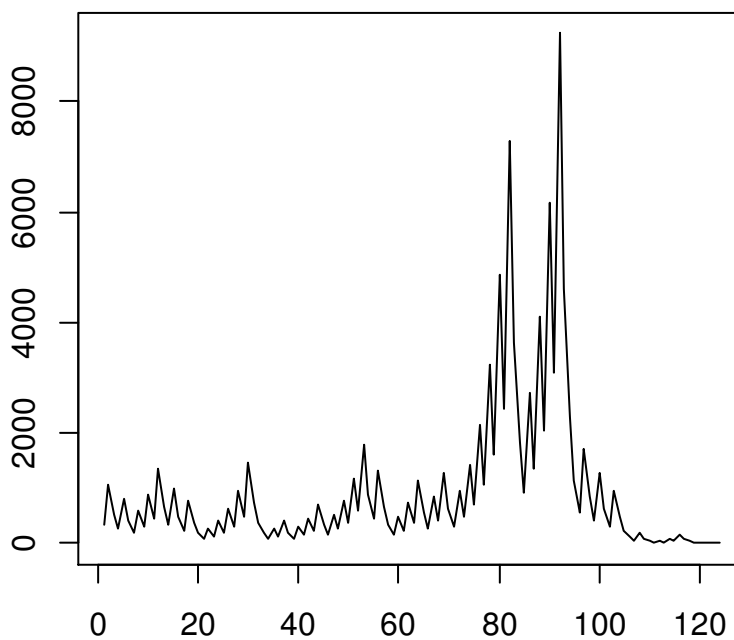
```
> collatz.report(3)
$sequence
[1] 3 10 5 16 8 4

$length
[1] 6

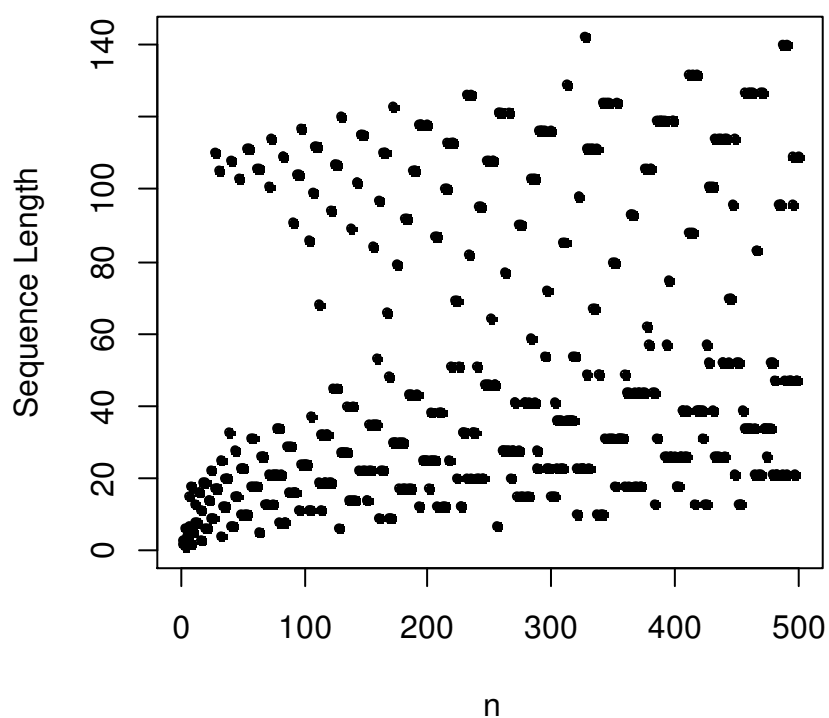
$maximum
[1] 16
```

אפשר ומומלץ להשתמש בפונקציה `collatz.report` בסעיפים הבאים.

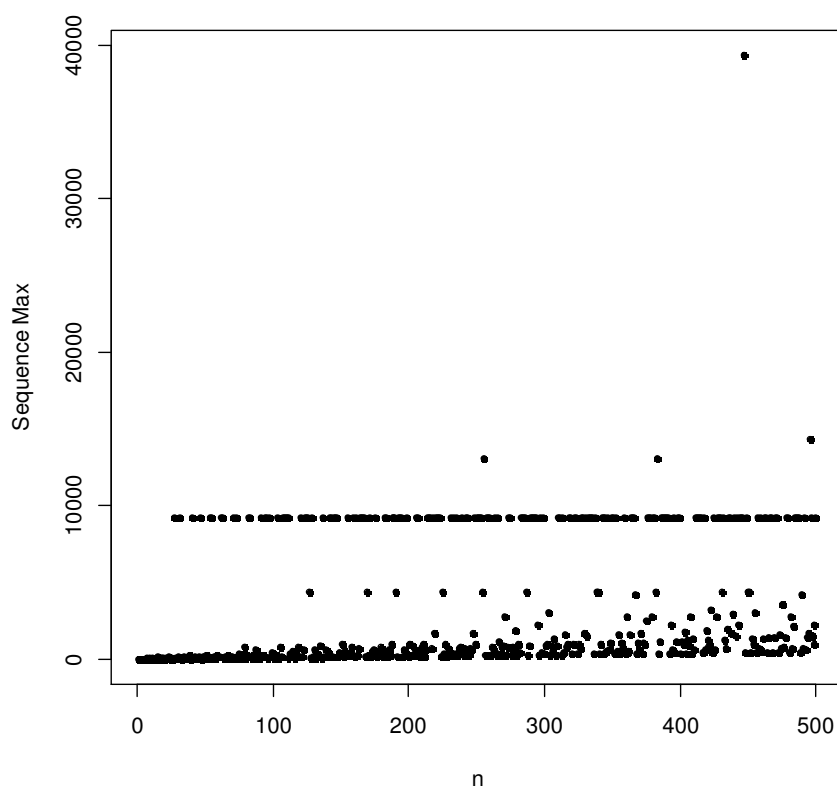
ב. כתבו פונקציה בשם `plot.collatz.seq` המקבלת כארגומנט מספר טבעי n , ויוצרת גרף של סדרת קולאץ' שמתחילה מ- n (ומסתיימת ב-4 הראשון). למשל, הפקודה `plot.collatz.seq(353)` תיצור את הגרף הבא:



ג. הכינו דיאגרמה המראה לכל מספר טבעי בין 1 ל-500 את אורך סדרת קולאץ' המתחילה ממנו.



ד. הכינו דיאגרמה המראה לכל מספר טבעי בין 1 ל-500 את האיבר המקסימלי בסדרת קולאץ' המתחילה ממנו.



6. בקובץ `inflation.unemployment.txt` (שבאתר הקורס) מופיעים נתונים אודות שיעור האינפלציה והאבטלה בשנים 1988-2000. כתבו סקריפט המציג את הנתונים הנ"ל בגרף באופן הבא:



7. סטטיסטיקה והסתברות

1. כתבו פונקציה בשם `my.summary` המקבלת כארגומנט וקטור מספרי, ומחזירה רשימה בת ארבעה פריטים:

- הפריט הראשון נקרא `mean`, והוא ממוצע הווקטור
- הפריט השני נקרא `median`, והוא החציון של הווקטור
- הפריט השלישי נקרא `var`, והוא השונות של הווקטור
- הפריט הרביעי נקרא `stdev`, והוא סטיית התקן של הווקטור

למשל,

```
> my.summary(c(3, 10, 1, 12, 5, 6))
$mean
[1] 6.166667

$median
[1] 5.5

$var
[1] 17.36667

$stdev
[1] 4.167333
```

2. נתון כי $X \sim \text{Pois}(6.1)$ ו- $Y \sim \text{Bin}(7, 0.4)$ הם בלתי-תלויים. חשבו את –

א. $P(3 \leq X \leq 7)$

ב. $P(X > 4)$

ג. $E\left(\frac{Y^2}{Y+1}\right)$

ד. $P(X = Y)$

3. מוציאים שני כדורים באקראי בלי החזרה מכד שבו 20 כדורים, הממוספרים בין 1 ל-20. נסמן ב- X את מכפלת שני המספרים שהתקבלו. חשבו את –

א. $P(X \geq 100)$

ב. $E(X)$

ג. $\text{Var}(X)$

4. כתבו פונקציה בשם f המקבלת כארגומנטים מספרים n, p , ו- k , ומחזירה את $E(X^k)$, כאשר X הוא משתנה מקרי $\text{Bin}(n, p)$.

5. תזכורת: אם X הוא משתנה מקרי בדיד המתפלג אחיד בין a ל- b (כאשר a ו- b הם מספרים שלמים), פונקציית ההסתברות ופונקציית ההתפלגות המצטברת שלו הן

$$P_X(x) = \begin{cases} \frac{1}{b-a+1}, & \text{שלם } x, \ a \leq x \leq b \\ 0 & \text{אחרת} \end{cases} \quad F_X(x) = \begin{cases} 0 & x < a \\ \frac{\lfloor x \rfloor - a + 1}{b - a + 1} & a \leq x \leq b \\ 1 & x \geq b \end{cases}$$

כתבו שתי פונקציות בשם `dunif.d` ו-`punif.d` הפועלות בדומה לפונקציות שראינו בהרצאה, עבור משתנה מקרי אחיד בדיד כנ"ל (ה-`d` בשמות הפונקציות מציין `discrete`, שזה "בדיד"). למשל,

```
> dunif.d(4, 2, 6)
[1] 0.2
> dunif.d(17, 2, 6)
[1] 0
> dunif.d(4.3, 2, 6)
[1] 0
> punif.d(4, 2, 6)
[1] 0.6
> punif.d(17, 2, 6)
[1] 1
> punif.d(4.3, 2, 6)
[1] 0.6
```

אין צורך שהפונקציות יידעו לטפל בווקטורים.

8. מספרים אקראיים וסימולציה

1. כתבו פונקציה (ללא ארגומנטים) בשם f , המגרילה מספר מקרי שהוא

- -3 בהסתברות 0.1
- 8.5 בהסתברות 0.7
- 17 בהסתברות 0.2

הפונקציה בפעולה:

```
> f()
[1] 17
> f()
[1] 8.5
> f()
[1] 8.5
```

2. נתון כי $X \sim \text{Pois}(6.1)$ ו- $Y \sim \text{Bin}(7, 0.4)$ הם בלתי-תלויים.

א. אמדו בסימולציה את $P(X > 8)$.

ב. אמדו בסימולציה את $E\left(\frac{X^2}{X+1}\right)$.

ג. אמדו בסימולציה את $P(X > 2Y)$.

ד. אמדו בסימולציה את $E\left(\frac{XY}{\sqrt{X}+1}\right)$.

3. בכד נמצאים 100 כדורים, הממוספרים מ-1 עד 100. אדם א' מוציא מהכד שלושה כדורים, ואדם ב' מוציא ממנו שני כדורים (שני האנשים מוציאים עם החזרה). מהי ההסתברות שהמספר המקסימלי מבין המספרים שהוציא אדם א', גדול מסכום המספרים שהוציא אדם ב'?

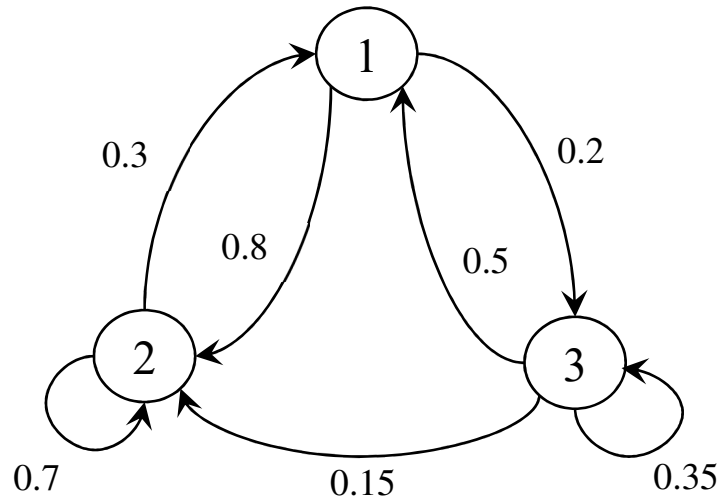
4. מוציאים שני כדורים באקראי בלי החזרה מכד שבו 20 כדורים, הממוספרים בין 1 ל-20. נסמן ב- X את מכפלת שני המספרים שהתקבלו. אמדו בסימולציה את –

א. $P(X \geq 100)$

ב. $E(X)$

ג. $\text{Var}(X)$

5. נתונה שרשרת מרקוב בעלת 3 מצבים :



א. כתבו פונקציה בשם next.step המקבלת כארגומנט את המצב הנוכחי בשרשרת, ומגרילה את המצב הבא.

אמדו באמצעות סימולציה את התשובות לשאלות הבאות.

- ב. אם מתחילים ממצב 1, מהי ההסתברות להימצא במצב 3 לאחר 10 צעדים?
 ג. אם מתחילים ממצב 1, מהי ההסתברות **לא לבקר כלל** במצב 3 במהלך 5 הצעדים הראשונים?
 ד. אם מתחילים ממצב 2, מהי תוחלת מספר הצעדים עד הביקור **השני** במצב 3?
 ה. נניח כי בכל הגעה למצב 1 מקבלים 10 נקודות, בכל הגעה למצב 2 מקבלים 15 נקודות, ובכל הגעה למצב 3 **משלמים** 25 נקודות. אם מתחילים ממצב 1 עם אפס נקודות, מהי ההסתברות שלאחר 5 צעדים תצטבר "יתרה" של יותר מ-50 נקודות?

6. מטילים קובייה הוגנת 10 פעמים. נסמן ב- X את מספר המספרים **השונים זה מזה** שהתקבלו (למשל, אם המספרים הם 2, 5, 4, 6, 2, 5, 4, 6, 5, 4, אז התקבלו ארבעה מספרים שונים זה מזה – 2, 4, 5, 6 – ולכן $X = 4$).

אמדו בסימולציה את –

- א. $P(X = 4)$.
 ב. $E(X)$.
 ג. החציון של X .
 ד. $\text{Var}(X)$.
 ה. שרטטו את פונקציית ההתפלגות המצטברת של X .

7. נתונים 10 כדורים הממוספרים 1-10, ו-10 קופסאות הממוספרות גם הן 1-10. מכניסים באקראי כדור אחד לכל קופסה. מהי ההסתברות שלפחות 2 כדורים יימצאו בקופסה עם המספר שלהם?

8. משחק בקזינו מתנהל באופן הבא.

- בשלב הראשון המשתתף מטיל קובייה. אם קיבל 3 או פחות, המשחק נגמר והמשתתף אינו זוכה בדבר; אם קיבל 4 או יותר, הוא עובר לשלב השני.
- בשלב השני, המשתתף מוציא x כדורים באקראי (ללא החזרה) מתוך כד ובו 10 כדורים הממוספרים מ-1 עד 10, כאשר x הוא המספר שקיבל בשלב הראשון.
- אם מכפלת המספרים שהתקבלו בשלב השני גדולה מ-100, המשתתף זוכה בסכום כספי שהוא המכפלה הני"ל, אחרת אינו זוכה בדבר.

מהי תוחלת סכום הזכייה במשחק?

9. מטילים קובייה הוגנת 10 פעמים. מהי ההסתברות שיתקבל לפחות זוג הטלות סמוכות אחד, שסכום המספרים בהן הוא 4 או פחות? כתבו סקריפט העונה על השאלה בסימולציה.

למשל, בסדרת המספרים הבאה יש זוג אחד של מספרים סמוכים שסכומם 4 או פחות:

1 5 6 1 2 5 1 6 4 3

בסדרה הבאה יש שני זוגות כאלה:

1 5 6 1 2 5 1 6 1 3

ובסדרה הזו אין אף זוג שכזה:

1 5 6 3 2 5 1 6 4 3

9. ביצועים ואופטימיזציה של קוד

1. א. כתבו פונקציה בשם f1 המקבלת כארגומנט שני וקטורים מספריים, x ו-y (לאו דווקא באותו האורך), ומחזירה מטריצה שבה האיבר בשורה i-ה ובעמודה j-ה הוא $(x_i + y_j)^2$.

למשל, אם x ו-y הם

```
> x
[1] 1 2 3
> y
[1] 2 0 4 1
```

אז

```
> f1(x, y)
      [,1] [,2] [,3] [,4]
[1,]     9     1    25     4
[2,]    16     4    36     9
[3,]    25     9    49    16
```

בפתרון בסעיף זה השתמשו בשתי לולאות מקוננות: אחת שעוברת על ערכי הווקטור x, ואחת על ערכי y.

ב. כתבו פונקציה בשם f2 העושה בדיוק מה ש-f1 עושה, אבל משתמשת בלולאה אחת בלבד.

ג. כתבו פונקציה בשם f3 העושה בדיוק מה ששתי הפונקציות הקודמות עושות, אבל לא משתמשת בלולאות בכלל, אלא בפונקציה outer. היעזרו בתיעוד של R כדי להבין מה הפונקציה outer עושה.

ד. כתבו פונקציה בשם f4 העושה בדיוק מה ששלוש הפונקציות הקודמות עושות, אבל לא משתמשת בלולאות או בפונקציה outer, אלא משתמשת ביכולת של R לעבוד באופן וקטורי ובכלל המיחזור ביצירת מטריצות.

ה. השוו את זמני הריצה של שלוש הפונקציות שכתבתם באמצעות microbenchmark, על הקלט לדוגמה מסעיף א'.

ו. חזרו על ההשוואה, כאשר הקלט הוא שרשור של x עם עצמו פעמיים (כלומר c(x, x)) וכנ"ל עם y. האם התמונה המתקבלת בהשוואה זהה לתמונה בסעיף הקודם?

2. בשאלה זו נחקור עוד את הבדלי הביצועים בין ארבע הפונקציות מהשאלה הקודמת, כתלות בגודל הקלט.

א. השוו את זמני הריצה של ארבע הפונקציות באמצעות `microbenchmark`, כאשר הקלט שלהן הוא שני וקטורים x ו- y באורך 15 שאבריהם הם מספרים שלמים אקראיים בתחום 1-10. במקום רק להריץ את `microbenchmark` ולראות את הפלט שהיא מדפיסה לקונסולה, הכניסו את הערך שהיא מחזירה לתוך משתנה בשם `mbm`, ואז הקלידו בקונסולה `mbm`, כך שתקבל על המסך טבלת ההשוואה המוכרת.

ב. כש-`microbenchmark` פעלה, היא הריצה 100 פעמים כל אחת מ-4 הפונקציות, כך שסך הכל היא קראה לפונקציות 400 פעמים. כדי שההשוואה תהיה יותר אמינה, סדר הקריאה לפונקציות הוא אקראי.

המשתנה `mbm` שיצרתם בסעיף א' הוא רשימה בעלת שני פריטים: הפריט הראשון נקרא `expr`, והוא פקטור באורך 400 שמכיל את סדר הקריאות לארבע הפונקציות; הפריט השני נקרא `time`, והוא וקטור באורך 400 שמכיל את זמני הריצה (בננו-שניות) של כל אחת מהקריאות לפונקציות.

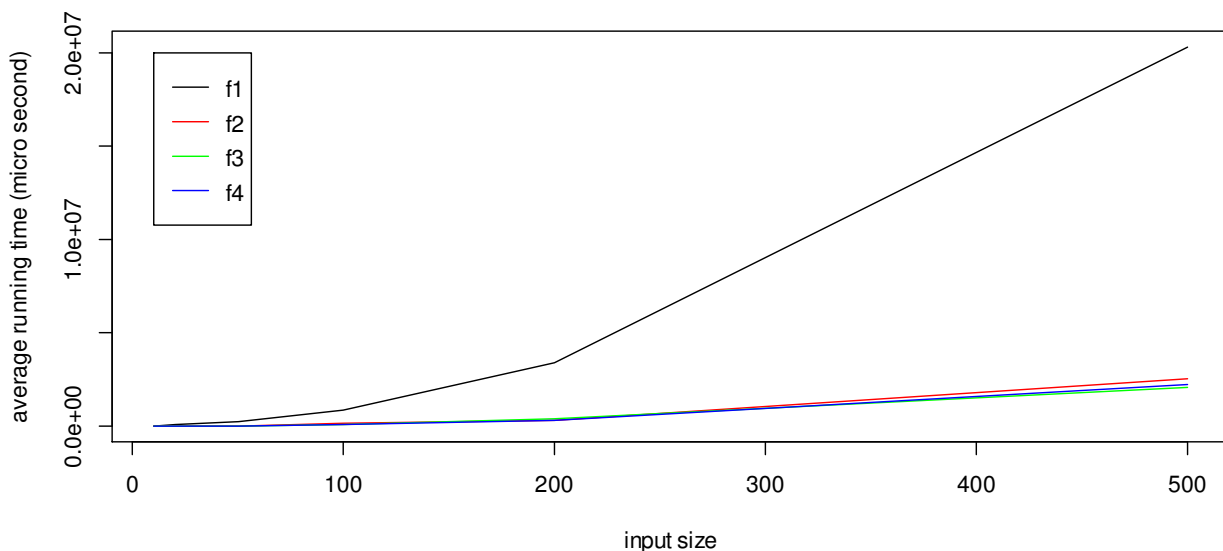
```
> head(mbm$expr)
[1] f2(x, y) f3(x, y) f2(x, y) f4(x, y) f4(x, y) f1(x, y)
Levels: f1(x, y) f2(x, y) f3(x, y) f4(x, y)
> head(mbm$time)
[1] 42200 23300 11900 12100 6800 35800
```

השתמשו בפונקציה `tapply` (בלי להשתמש בלולאות) כדי ליצור מהמשתנה `mbm` וקטור באורך 4 המכיל את ממוצע זמן הריצה של כל אחת מארבע הפונקציות. שימו לב: בגלל שהיחידות של `mbm` הן תמיד ננו-שניות, בעוד היחידות שמודפסות למסך הן לפעמים ננו-שניות, לפעמים מיקרו-שניות וכו', ייתכנו הבדלים של פי אלף, פי מיליון, וכו' בין התוצאות בשתי השיטות.

ג. כעת השוו באמצעות `microbenchmark` את ממוצע זמני הריצה של ארבע הפונקציות על קלט באורך הולך וגדל: בהשוואה הראשונה האורך של x ו- y הוא 10, בשנייה 20, ואז 50, 100, 200, ו-500 (סה"כ 6 השוואות). בכל המקרים הווקטורים צריכים להכיל מספרים שלמים אקראיים בתחום 1-10. רכזו את תוצאות ההשוואה במטריצה מספרית בשם `time.averages`, שהיא מטריצה עם 6 שורות ו-4 עמודות. שמות העמודות של המטריצה צריכים להיות `f1`, `f2`, `f3`, `f4`:

```
> time.averages
      f1      f2      f3      f4
[1,] 12944   6546   6451   4782
[2,] 39331  12132   8124   6473
[3,] 209916  37317  20032  17632
[4,] 881043 159688  86961  77931
[5,] 3368582 320950  372557 331157
[6,] 20296298 2551768 2099499 2255519
```

ד. הכינו גרף המתאר את זמן הריצה כפונקציה של אורך הקלט עבור ארבע הפונקציות:



10. עיבוד נתונים עם dplyr

1. השאלות הבאות מתייחסות לטבלה `flights` מהחבילה `nycflights13`.

- א. כתבו פקודה המוצאת את הפרטים המלאים של כל הטיסות שהמריאו בחודש פברואר.
 - ב. כתבו פקודה המוצאת את הפרטים המלאים של כל הטיסות שהמריאו בחודש פברואר מ-JFK.
 - ג. כתבו פקודה המוצאת את היום, שעת ההמראה, ומספר הטיסה של כל הטיסות שהמריאו בחודש פברואר.
 - ד. כתבו פקודה המוצאת את רשימת כל היעדים **השונים** שאליהם טסו.
 - ה. כתבו פקודה המוצאת את מספר היעדים השונים שאליהם טסו.
 - ו. כתבו פקודה המוצאת את מספר היעדים השונים שאליהם טסה חברת United Airlines (UA).
 - ז. כתבו פקודה המוצאת את פרטי הטיסה שהמריאה באיחור הגדול ביותר.
 - ח. כתבו פקודה המוצאת את המרחק של הטיסה שהכי **הקדימה** להמריא.
 - ט. כתבו פקודה שמחזירה טבלה חדשה הזוהה לטבלה `flights`, אבל יש בה עמודה נוספת בשם `duration` שבה כתוב "short" אם זמן הטיסה היה שעותיים או פחות, ו-"long" אחרת.
 - י. כתבו פקודה המוצאת את מרחק הטיסה הממוצע עבור כל חברת תעופה.
- החבילה `nycflights13` מכילה גם את הטבלה `airports`, עם מידע אודות 1,458 שדות תעופה בארה"ב.
- י"א. כתבו פקודה המוצאת את פרטי שדות התעופה שהגובה שלהם מעל פני הים (alt) הוא מעל 1,000 רגל.
 - י"ב. כתבו פקודה המוצאת את השמות (faa) של שדות התעופה שהגובה שלהם מעל פני הים הוא מעל 1,000 רגל.
 - י"ג. כתבו פקודה המוצאת את רשימת חברות התעופה השונות שטסו לשדות תעופה שהגובה שלהם מעל פני הים הוא מעל 1,000 רגל.

11. אובייקטים ב-R

1. בהרצאה למדנו שהטווח המספרי של משתנים מטיפוס integer (מספרים שלמים) ב-R הוא מוגבל יחסית – בערך בין מינוס 2 מיליארד לפלוס 2 מיליארד. בתרגיל זה נכתוב פונקציה בשם `add.two.integers` שתאפשר לבצע פעולת חיבור בין שני מספרים שלמים הרבה יותר גדולים.

בייצוג עשרוני, המספר 2 מיליארד הוא 2000000000 – כלומר יש לו 10 ספרות. כדי שנוכל לעבוד עם מספרים בעלי יותר ספרות (אפילו אלפי ספרות), נייצג את המספרים בקלט של הפונקציה **כמחרוזות**, וגם הפלט של הפונקציה יהיה מחרוזת. למשל,

```
> add.two.integers("25", "300")
[1] "325"
> add.two.integers("12345678901234567890", "1717171717171717")
[1] "12347396072951739607"
```

הפונקציה צריכה לעבוד באופן הבא :

- להמיר את המחרוזות לווקטורים מספריים, כך שכל איבר בווקטור יהיה ספרה.
- לבצע את פעולת החיבור בין שני המספרים באופן שלומדים לעשות בבית ספר יסודי (מחברים כל פעם זוג ספרות החל מספרת האחדות, עם העברה של ספרת העשרות של התוצאה לחיבור זוג הספרות הבא), ולקבל וקטור מספרי שהאיברים שלו הם הספרות של התוצאה. מותר להשתמש בלולאה.
- להמיר את וקטור התוצאה למחרוזת, ולהחזיר אותה.

2. א. כתבו פונקציה בשם `add.integers` המקבלת כארגומנט וקטור מחרוזות המייצגות מספרים שלמים, ומחזירה מחרוזת של הסכום שלהן. למשל,

```
> add.integers(c("17", "3", "500"))
[1] "520"
> add.integers("17")
[1] "17"
```

הפונקציה צריכה להתבסס על הפונקציה `add.two.integers` מהשאלה הקודמת, ומותר להשתמש בלולאה.

ב. חזרו על סעיף א', אך הפעם השתמשו ברקורסיה במקום בלולאה.