

LESSON 12

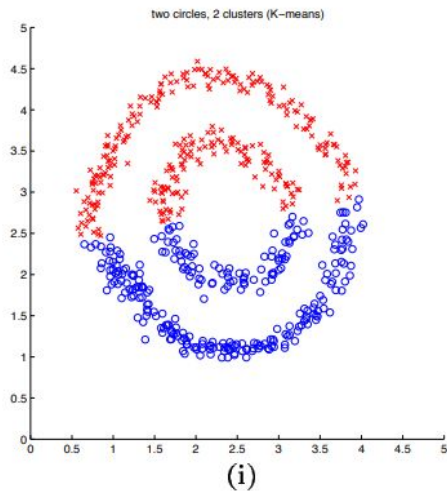
Final Project, Python

GUIDELINES

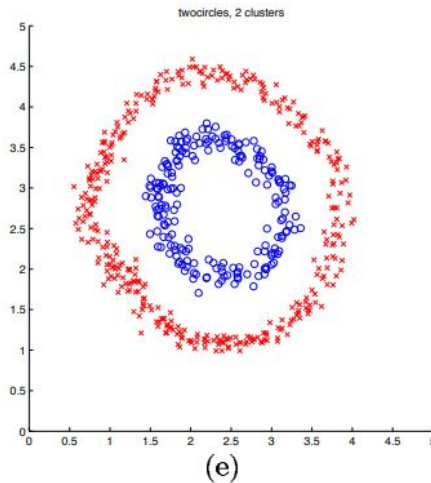
- Submission Date: 24/04/2022
- NO EXTENSION!!!

MOTIVATION - SPECTRAL CLUSTERING

K-means



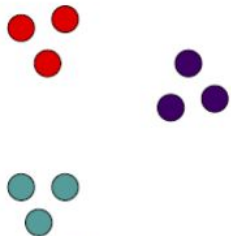
Spectral clustering



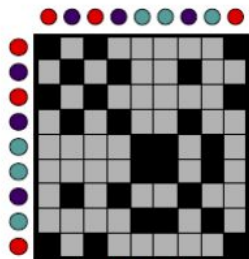
[Shi & Malik '00; Ng, Jordan, Weiss NIPS '01]

FROM DATA POINTS TO GRAPH

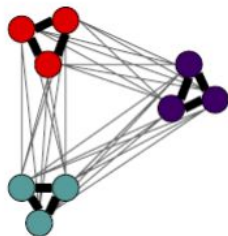
- Given: d-dimensional points: x_1, x_2, \dots, x_n .
- Transform them into a graph (Similarity Graph).
 - $G = (V, E; W)$ - undirected with no self loops.



Data



Similarities



Similarity graph

WEIGHTED ADJACENCY MATRIX

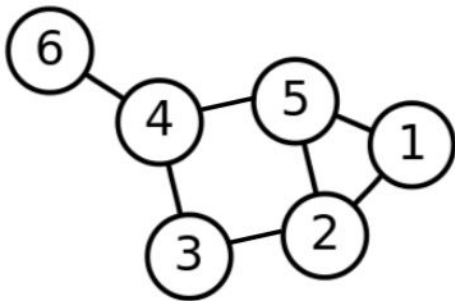
- Gaussian RBF

$$w_{ij} = \exp\left[-\frac{(x_i - x_j)^2}{2}\right]$$

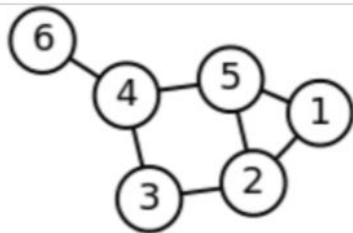
- W is symmetric, non-negative and no self loops($W_{ii}=0$)

EXAMPLE

- For simplicity, in the next example we will show a non fully connected graph, with all weights set to 1
- We are given d-dimensional data points: $x_1 \dots x_n$
- Choose random points and connect them, and we get:



GRAPH NOTATIONS



D (diagonal) degree matrix

$$D_{ii} = \sum_{j=1}^n w_{ij} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

W weight matrix

$$W = (w_{ij}) \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

L graph Laplacian

$$L = D - W$$

$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

L_{norm} normalized graph Laplacian

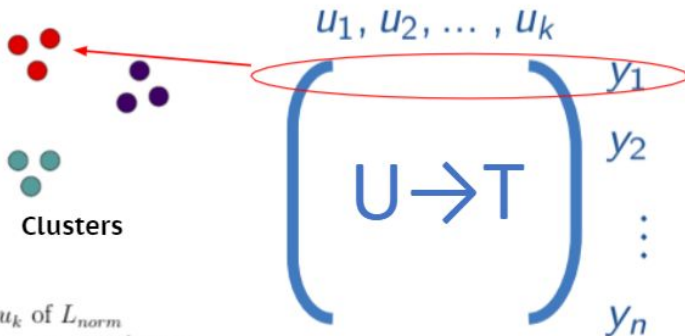
$$L_{norm} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$\begin{pmatrix} 1 & -\frac{1}{\sqrt{6}} & 0 & 0 & -\frac{1}{\sqrt{6}} & 0 \\ -\frac{1}{\sqrt{6}} & 1 & -\frac{1}{\sqrt{6}} & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{\sqrt{6}} & 1 & -\frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{6}} & 1 & -\frac{1}{3} & -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{\sqrt{3}} & 0 & 1 \end{pmatrix}$$

NORMALIZED SPECTRAL CLUSTERING (NG, JORDAN, AND WEISS)

Given n points

$$X = \{x_1, x_2, \dots, x_n\}$$



Clusters

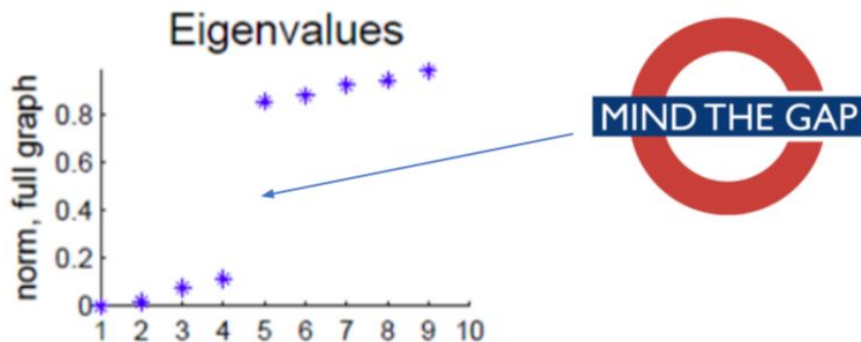
- 1: Form the weighted adjacency matrix W from X
- 2: Compute the normalized graph Laplacian L_{norm}
- 3: Determine k and obtain the first k eigenvectors u_1, \dots, u_k of L_{norm}
- 4: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns
- 5: Form the matrix $T \in \mathbb{R}^{n \times k}$ from U by renormalizing each of U 's rows to have unit length, that is set $t_{ij} = u_{ij} / (\sum_j u_{ij}^2)^{1/2}$
- 6: Treating each row of T as a point in \mathbb{R}^k , cluster them into k clusters via the K-means algorithm
- 7: Assign the original point x_i to cluster j if and only if row i of the matrix T was assigned to cluster j

DETERMINE K ?

- K-Means and Spectral Clustering require k (the number of clusters) as an input
- Possible solutions:
 - Prior knowledge about the data (e.g. as user input)
 - Eigengap Heuristic

EIGENGAP HEURISTIC

- Sort $0 \leq \lambda_1 \leq \dots \leq \lambda_n$
- Delta: $\delta_i = |\lambda_i - \lambda_{i+1}|$
- Gap: $k = \operatorname{argmax}_i(\delta_i), \quad i = 1, \dots, \frac{n}{2}$



FINDING EIGENVALUES AND EIGENVECTORS - JACOBI ALGORITHM

(a) Build a rotation matrix P (as explained below).

(b) Transform the matrix A to:

$$A' = P^T A P$$

(c) Repeat a,b until A' is diagonal matrix.

(d) The diagonal of final A' is the eigenvalues of A .

(e) Calculate eigenvectors of A by multiplying all the rotation matrices:

$$V = P_1 P_2 P_3 \dots$$

ROTATION MATRIX P

Let S be a symmetric matrix, and P is Jacobi rotation matrix of the form:

$$P = \begin{pmatrix} 1 & & & & \\ & \dots & & & \\ & & c & \dots & s \\ & & \vdots & 1 & \vdots \\ & & -s & \dots & c \\ & & & & \dots & 1 \end{pmatrix}$$

JACOBI TERMINOLOGY

Pivot

The A_{ij} is the off-diagonal element with the largest absolute value.

Obtain c, t

$$\theta = \cot 2\phi = \frac{A_{jj} - A_{ii}}{2A_{ij}}$$

$$t = \frac{\text{sign}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}$$

$$c = \frac{1}{\sqrt{t^2 + 1}}, \quad s = tc$$

Note: We define $\text{sign}(0) = 1$

UPDATE A-MATRIX EFFICIENTLY

After each transformation in step 2, the changed elements of A are only the i and j rows and columns. Therefore, using the symmetry of A we can obtain the following formula to calculate A' :

$$a'_{ri} = ca_{ri} - sa_{rj} \quad r \neq i, j$$

$$a'_{rj} = ca_{rj} + sa_{ri} \quad r \neq i, j$$

$$a'_{ii} = c^2 a_{ii} + s^2 a_{jj} - 2sca_{ij}$$

$$a'_{jj} = s^2 a_{ii} + c^2 a_{jj} + 2sca_{ij}$$

$$a'_{ij} = (c^2 - s^2)a_{ij} + sc(a_{ii} - a_{jj}) \Rightarrow a'_{ij} = 0$$

Note: A' is always symmetric.

JACOBI EXAMPLE

- Build a rotation matrix P (as explained below).
- Transform the matrix A to:

$$A' = P^T A P$$
- Repeat a,b until A' is diagonal matrix.
- The diagonal of final A' is the eigenvalues of A .
- Calculate eigenvectors of A by multiplying all the rotation matrices:

$$V = P_1 P_2 P_3 \dots$$

Let S be a symmetric matrix, and P is Jacobi rotation matrix of the form:

$$P = \begin{pmatrix} 1 & & & & \\ & \dots & & & \\ & & c & \dots & s \\ & & \vdots & 1 & \vdots \\ & & -s & \dots & c \\ & & & & \dots & 1 \end{pmatrix}$$

Pivot

The A_{ij} is the off-diagonal element with **the largest absolute value**.

Obtain c, t

$$\theta = \cot 2\phi = \frac{A_{jj} - A_{ii}}{2A_{ij}}$$

$$t = \frac{\text{sign}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}$$

$$c = \frac{1}{\sqrt{t^2 + 1}}, \quad s = tc$$

Note: We define $\text{sign}(0) = 1$

$$a'_{ri} = ca_{ri} - sa_{rj} \quad r \neq i, j$$

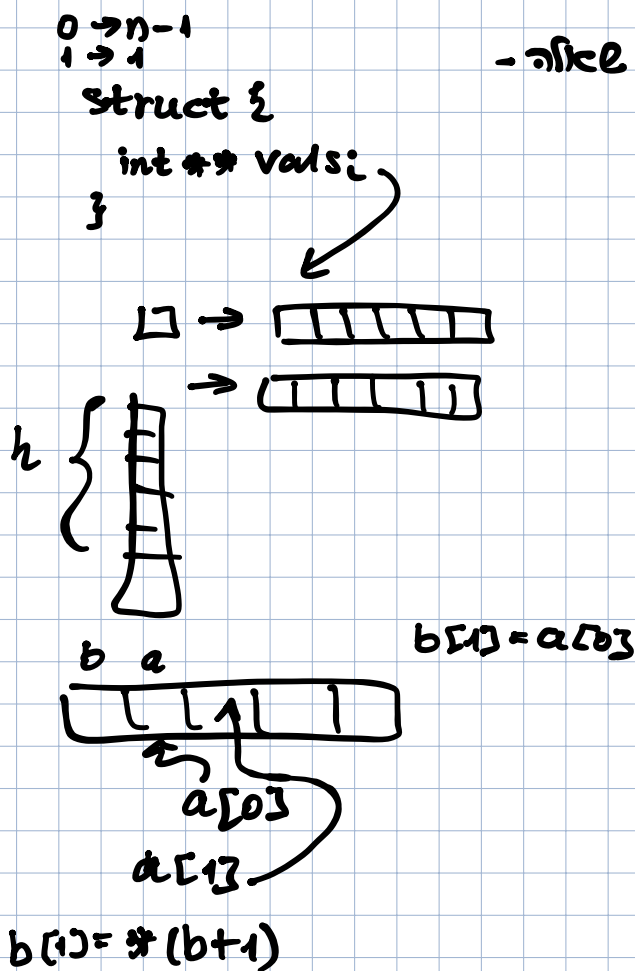
$$a'_{rj} = ca_{rj} + sa_{ri} \quad r \neq i, j$$

$$a'_{ii} = c^2 a_{ii} + s^2 a_{jj} - 2sca_{ij}$$

$$a'_{jj} = s^2 a_{ii} + c^2 a_{jj} + 2sca_{ij}$$

$$a'_{ij} = (c^2 - s^2)a_{ij} + sc(a_{ii} - a_{jj}) \Rightarrow a'_{ij} = 0$$

EXAM!



- עם `malloc` ולתת `free`.
- עם משתנים אחר כך בסונקציה וזהם `free`.
- לאזהיט י: פונקציות אסיון עם `free`.
- לתקן בתמונה לציאת: `result → vals[i]`.
- `result → vals--i`
- בשביל לתמוך במסלול פונקציות
- יהיו בין 1 ל- n .

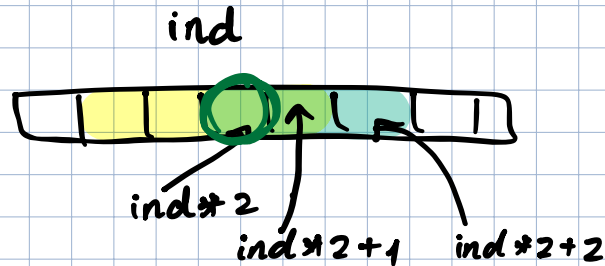
שאלה 3

כוננים למצב איפה חידים עלויות סנכרון כפי שלא תהיה בעיה.

פתרון

בין 2 ל-3 או בין שורה 3 ל-4. צריך לחכות שכל יציא אנקודה מסיימת

ויסימו.



בעיות הסנכרון יהיו רק עם האינדיקס. יש בעיה עם הסנכרון כאשר במורה 3

$ind-1$ לא סיים ויש בעיה בחסיפה הראשונה.

שאלה 4

$$a = 30 = \frac{180}{G(2)}$$

$$b = 15 = \frac{180}{G(3)}$$

$$G(x) - 6 \begin{cases} G(2) = 6 \\ G(3) = 12 \end{cases}$$

#define G(x) (x*(x+1)) צריך לעשות:

$$x = ? : (x = 3)? \dots$$

$G(x * (x+1))$ כמה שמים סוגריים?

$$G(y * y) = y + \underbrace{(y+1) * y}_{\text{Ⓢ}} + (y+1)$$

זה מתבצע ראשון לבי
החוקים זה לא מה שרצינו

$$G(q++) \rightarrow \underbrace{(x * x)}_{q++} + 1 \quad \text{צבאם לבדף:}$$

++ q ++ -- z הצבר הזה לא מוזכר:

שאלה 1

אילו היתר - אולי אפשר ליצור מערך בגודל מסוים.

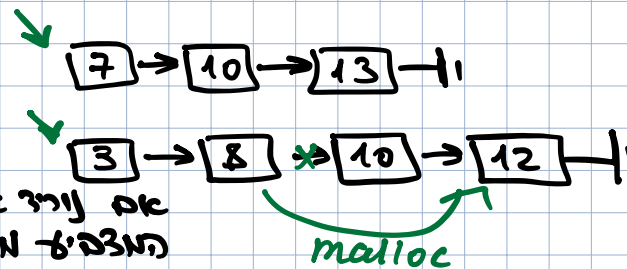
a	b	c	d	/0
---	---	---	---	----

1	2	3	4	/0
---	---	---	---	----

a	1	b	2	c	3	d	4	/0
---	---	---	---	---	---	---	---	----

• מחזרת חייבת להסתיים ב-0 / כי למשל printf מחפש /0 כסיום או לציין זה צ"ל כי כמה איברים יש במחזרת.

ב. יש שתי נשמות לא ריקות ואיברים שממוינים בסדר שלה.

צמחה

אם נוצר את [3]
המזכיר משהיה ל-8.
נראה שהשני יהיה גם מחול לפונקציה.
אחרת אחרי מזכיר יבדל למטהו
בסופו רקר לציין. בגוף זה שמים
במשתנים List** ולא List*.

המחרוזת ממוינת ולכן צריך להתקדם לזיכרון המזכיר

לחפש שנייה במקרה. אפשרות אחרת היא לקדם רק את המזכיר לאיבר

שקטן יותר עד שיהיה שווה ולסיים כשמזכיר אסוף. הפונקציה צריכה לחזיר

כמה איברים חרצו.

החלטה תתבצע כל עוד שניהם לא הגיעו ל-NULL. ברגע שאחד מהם מסתיים, סיימנו.

יש 3 אפשרויות לעקר שלמות הם מזכירים הראשון קטן, שניה מהפני או צדדים.

• אפשר וצריך לשחרר free ו-strud.

צריך לדאוג לזה שתהיה לנו הזכירה טובה. מאתחלים את המזכיר שלו ל-NULL

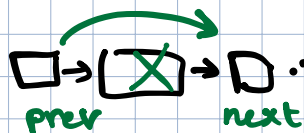
כי אם מוחקים את כולם אין בעיה. האיבר הראשון שאינו NULL, מזכירים אליו בתור

התחלה.

int** a a = 3
→ *a 3 a
 3
*(110) = 3

Stack - ויזה ב.

סהר הרשימה הפנייה שמורים מצביע לאיבר ה- $prev$ כי יכול להיות שהוא

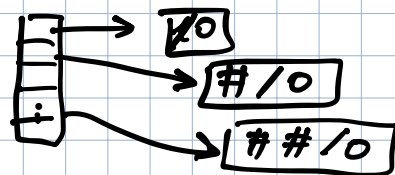
ימחק. שהוא נמחק, צריך לעשות: $prev.next = this.next$ ולעשה. 

אכן קוצב שמורים את ה- $prev$ ורק אז מקבלים.

- כש שניין בין האיברים הנוכחיים של שתי הרשימות צריך להתקדם בשניהם, מוסיפים ה- $counter$, שמורים את ההצבה עצמה ומשחרים אותה אחרי שקיצענו. אם ה- $prev$ לא מלא לעשה את המעקף.

כדי להיות בטוח רצו יכול להיות שבספסי מקני קצה. במקרה כזה לוקחים צומח במטה ומנסים להציב על מקרים כללים ואז מקני קצה שכבא לבזוק: השיטה מתאפסת, האיבר הראשון נמחק... אצבאג יצירת וחתקו.

ג. יש מ מחיצות. זה שמור מעדך כך:



- יש $struct$ שמכיל מצביע נפלא. מחצרים מצביע אפיו ולכן צריכים גם להקצות לו זיכרון. אבל אם אין מצביע יש הוצה: המעריך מועתק ו- $return$ by value. זה יכול לתפוס את היצורין אם יש המעריך הוטמן איברים. כשמומים מצביע מועתקים רק 8 בתים.
 - אישם לה נשוצרים את המחיצות אישם $(i+1)$ בשביל חסיון 0.

שאלה 3

אישם לה שהתוכנית מצטיסה מ ללה מ \therefore

נצפה ש: $f(x) = (x-1)!$ נכזה בשורות 1-5 יעשו את זה. יש כאן

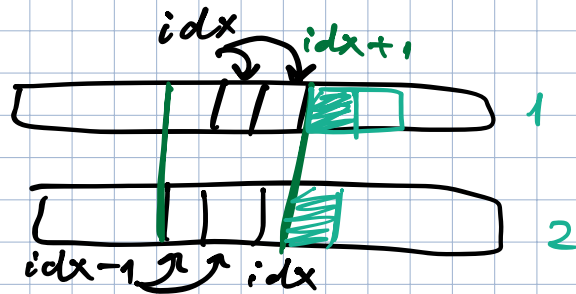
סוג של רקורסיה שחיה להיות לה בסיס. לכן צריך להקטין את x ב-1 או

משהו כזה לקבצק קריאה רקורסיה.

$define f(x) = f(x-1) \cdot x$ (אם $x > 1$)

פתרון אגשי:

$int f(int x)$
 אסור אישם פה $x-1$
 כי ה- $processor$
 היה משתגע
 $return f(x-1);$
 $\}$



על 2 כ:

יהיו בעיות סנכרון נשמעו מסתובב בטווח idx - Scope של הסק שלו.

מהתוצאה סטטית יש לנו הנה נשמעו מסתובב את התפקידים של idx ו- $idx+1$.

אכן נשי סנכרון בין השמות 13, 14. דואר צרכים אסימ ביקר אחת אחת

מה- thread משך זרימה הקטה לפני שהאחרים סימ.

נצל את הקד ידנית ונצל ארבעה שהספק צב ממנית את המודק.

אם נסתכל על idx סטצס, הוא מנין קטורה 8 שני אלמנטים: זילג בשוואה

היניהם ומחול. כל פה מחולק בין שני איקניו וממין: brick sort.

מק החיצה שלו (מס), זה צומח ל- bubble sort אבל הוא מתקצר בזמן (מס).

STRUCTURE

- 2 hours
- 1 paper formula sheet
- 4 questions:
 - 1 x open question in C
 - 2 x short answer (fill missing code) Python
 - 1 x short Eli's material
- Python Material:
 - Everything we learned in lectures
 - Data science oriented Python questions
 - https://www.practicaldatascience.org/html/class_schedule.html

שאלה 1

נתון כמות הייצור השנתית של 5 עובדים לאורך שנים (בסדר עולה) בטבלת PRODUCTION. מעוניינים לאמוד את התפוקה של עובדים חדשים NEW_EMPS (מכילה עמודת ID, VETEK), לפי הקריטריון הבא: תפוקה צפויה = ותק*אלפא, כך שאלפא זה ממוצע התפוקה של עובד 3 ו4 בחמשת השני הראשונות שלהם. השלם את השורה כדי לתת תחזית לתפוקת העובדים החדשים.

```
>>> production.shape
(20,5)
# -----
>>> print(new_emps['production'])
42105
265455
333008
...
```

שאלה 1

נתון כמות הייצור השנתית של 5 עובדים לאורך שנים (בסדר עולה) בטבלת PRODUCTION. מעוניינים לאמוד את התפוקה של עובדים חדשים NEW_EMPS (מכילה עמודת ID, VETEK), לפי הקריטריון הבא: תפוקה צפויה = ותק*אלפא, כך שאלפא זה ממוצע התפוקה של עובד 3 ו-4 בחמשת השני הראשונות שלהם. השלם את השורה כדי לתת תחזית לתפוקת העובדים החדשים.

```
>>> production.shape
(20,5)
>>> new_emps['production'] = new_emps['vetek']*production[:5,[2,3]].mean()
>>> print(new_emps['production'])
42105
265455
333008
...
```


שאלה 2

For each **continent** show the **continent** and number of countries with populations of at least 10 million.

```
import numpy as np
import pandas as pd
```

```
big_countries = world[world['population']>=10000000]
```

```
-----
```

```
print(big_per_cont)
```

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	188681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000
...				

שאלה 2

For each **continent** show the **continent** and number of countries with populations of at least 10 million.

```
import numpy as np
import pandas as pd

big_countries = world[world['population']>=10000000]
big_per_cont = big_countries.groupby(['name']).name.count()
print(big_per_cont)
```

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	188681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000
...				

שאלה 3

נתון מודל מאומן של סיווג על ידי רגרסיה לוגיסטית `model`, ברצוננו לבצע סיווג לנקודה $(-0.79415228, 2.10495117)$, השלם את החלק החסר:

```
# example of making a single class prediction
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import make_blobs
# generate 2d classification dataset
X, y = make_blobs(n_samples=100, centers=2, n_features=2, random_state=1)
```

שאלה 3

נתון מודל מאומן של סיווג על ידי רגרסיה לוגיסטית `model`, ברצוננו לבצע סיווג לנקודה: $(-0.794, 2.104)$, השלם את החלק החסר:

```
# example of making a single class prediction
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import make_blobs
# generate 2d classification dataset
X, y = make_blobs(n_samples=100, centers=2, n_features=2, random_state=1)

Xnew = [[-0.794, 2.104]]
model.predict(Xnew)
```