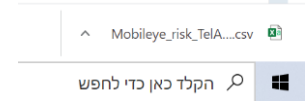


מגיש : ערד קוצר
ת"ז: 311226203
מסלול: מדמ"ח יזמות

חלק 1

שאלה 1



שאלה 2

Sklearn היא ספריית למידת מכונה אשר הוקמה בשנת 2007 על ידי דייוויד קורנפאו כפרויקט קיץ של גוגל ומאז תרמו אליה מתנדבים רבים כקוד פתוח. כיום הוא מתוחזק על ידי צוות מתנדבים כקוד פתוח. הספרייה עובדת עבור שפת התכנות Python 3 והיא כוללת אלגוריתמים שונים של סיווג, רגרסיה ואשכולות, כולל יערות אקראיים, הגדלת שיפוע k-means, DBSCAN, והיא נועדה לשתף פעולה עם הספריות המדעיות של Python NumPy ו- SciPy.

```
In [1]: #num 2
import sklearn
```

שאלה 3

```
In [2]: #num 3

import pandas as pd
import matplotlib.pyplot as plt
import geopy
```

שאלה 4

```
In [3]: #num 4
def read_data_from_csv(file_name):
    return pd.read_csv(file_name);

df = read_data_from_csv("Mobileye_risk_TelAviv_New.csv")
df.describe()
```

Out[3]:

	start_lat	start_long	end_lat	end_long	detection_drives_count	avg_speed	near_miss_pedestrian_ratio	near_miss_bicycle_ratio	near_
count	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	
mean	32.065733	34.775065	32.065730	34.775070	557.363048	34.631136	0.004319	0.000930	
std	0.019065	0.010403	0.019061	0.010399	1303.459900	13.204298	0.015200	0.004901	
min	32.029168	34.742794	32.029091	34.742794	50.000000	9.923498	0.000000	0.000000	
25%	32.050809	34.769196	32.050817	34.769217	93.000000	25.997728	0.000000	0.000000	
50%	32.064847	34.776714	32.064841	34.776717	188.000000	31.902938	0.000000	0.000000	
75%	32.078517	34.783262	32.078487	34.783254	481.000000	39.849982	0.000000	0.000000	
max	32.129996	34.791792	32.129996	34.791854	25047.000000	90.656510	0.476190	0.157895	

חלק 2:

שאלה 5

```
In [14]: from geopy.distance import geodesic
def distance(row):
    address1 = (row['start_lat'], row['start_long'])
    address2 = (row['end_lat'], row['end_long'])
    return (geodesic(address1, address2).m) #in meters

In [16]: df['section_length'] = df.apply(lambda x: distance(x), axis=1)

In [17]: df.head()

Out[17]:
```

io	avg_pedestrian_on_road_volume	avg_bicycle_on_road_volume	braking_count	cornering_count	harsh_braking_ratio	harsh_cornering_ratio	section_length
0	0.033557	0.033557	21	183	0.0	0.004966	10.010314
0	0.071429	0.015873	0	0	0.0	0.000000	24.101012
0	0.010327	0.000000	60	389	0.0	0.002571	9.673643
0	0.013605	0.003401	29	201	0.0	0.003253	10.088931
0	0.026667	0.000000	11	48	0.0	0.020833	71.514146

שאלה 6

```
In [19]: df.section_length.describe()

Out[19]: count    9266.000000
         mean      42.657029
         std       44.375733
         min        0.671996
         25%       11.204812
         50%       26.910230
         75%       62.054658
         max       449.794289
         Name: section_length, dtype: float64
```

הסבר – ניתן לראות שהקטע הארוך ביותר הוא ברוקן 449.794 מטר, בעוד שהכי קטן הוא באורך 0.6719 מטר. כמו כן המרחק הממוצע הוא 42.657 מטר למקטע.

שאלה 7

```
In [10]: def time(row):
         avg_time_hour = ((row['section_length'])/(row['avg_speed'])*1000)
         return (avg_time_hour)

In [11]: df['avg_time'] = df.apply(lambda x: time(x), axis=1)

In [12]: df.avg_time.describe()

Out[12]: count    9266.000000
         mean    1448.854185
         std    1687.832432
         min     24.307915
         25%    322.428115
         50%    736.008088
         75%    2093.566885
         max    17109.670249
         Name: avg_time, dtype: float64

In [13]: df['avg_time_sec'] = df.apply(lambda x: (time(x)*3600), axis=1)
```

שאלה 8

```
In [19]: df1 = df.drop(['start_lat', 'start_long', 'end_lat', 'end_long'], axis =1)
```

Out[17]:

```
In [17]: df1.describe()
```

	detection_drives_count	avg_speed	near_miss_pedestrian_ratio	near_miss_bicycle_ratio	near_miss_vehicle_ratio	avg_pedestrian_on_road_volume	avg_bicycle_on_road_volume	braking_count	cornering_count	harsh_braking_ratio	harsh_cornering_ratio	section_length	avg_time	avg_time_sec
count	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9266.000000	9.266000e+03
mean	557.363048	34.631136	0.004319	0.000930	0.008162	0.050953	0.026391	66.247788	276.613857	0.000002	0.015083	42.657029	1448.854185	5.215675e+06
std	1303.459900	13.204298	0.015200	0.004901	0.017387	0.074877	0.043735	138.458602	614.604640	0.000135	0.037954	44.375733	1687.832432	6.076197e+06
min	50.000000	9.923498	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.071996	24.307915	8.750848e+04
25%	93.000000	25.997728	0.000000	0.000000	0.000000	0.001731	0.000000	5.000000	38.000000	0.000000	0.000000	11.204812	322.428115	1.160741e+06
50%	188.000000	31.902938	0.000000	0.000000	0.000000	0.020090	0.010441	22.000000	87.000000	0.000000	0.001331	26.910230	736.008088	2.649629e+06
75%	481.000000	39.849982	0.000000	0.000000	0.009326	0.067423	0.033782	68.000000	265.000000	0.000000	0.011058	62.054658	2093.566885	7.536841e+06
max	25047.000000	90.656510	0.476190	0.157895	0.235294	0.780000	0.716418	2715.000000	7130.000000	0.012348	0.636364	449.794289	17109.670249	6.159481e+07

הסבר - העמדות של DATA המשתנים כול לא המשתנים שצריכים לעבור נרמול. נרמול. הם: 'detection_drives_count', 'avg_speed', 'avg_pedestrian_on_road_volume', 'avg_bicycle_on_road_volume', 'braking_count', 'cornering_count', 'section_length', 'avg_time'. צריכות לעבור נרמול וזאת משום שהערכים שלהם לא בין 0-1 כפי שאנחנו רוצים אותם. המשתנים שלא צריך לנרמל הן: 'near_miss_pedestrian_ratio', 'near_miss_bicycle_ratio', 'near_miss_vehicle_ratio', 'harsh_braking_ratio', 'harsh_cornering_ratio'.

שאלה 10

```
In [56]: from sklearn.preprocessing import MinMaxScaler

column_to_normalize = ['detection_drives_count', 'avg_speed', 'avg_pedestrian_on_road_volume', 'avg_bicycle_on_road_volume',
                        'braking_count', 'cornering_count', 'section_length', 'avg_time']

scaler = MinMaxScaler()

x_scaled = scaler.fit_transform(df1[column_to_normalize])

X_normalize = pd.DataFrame(x_scaled, columns=column_to_normalize, index=df1.index)

X_normalize1 = pd.concat([X_normalize, df2], axis=1)
pd.DataFrame(X_normalize1)
```

Out[56]:

	detection_drives_count	avg_speed	avg_pedestrian_on_road_volume	avg_bicycle_on_road_volume	braking_count	cornering_count	section_length	avg_time
0	0.009921	0.249457	0.043022	0.046840	0.007735	0.025666	0.020792	0
1	0.003040	0.201741	0.091575	0.022156	0.000000	0.000000	0.052166	0
2	0.021243	0.279347	0.013240	0.000000	0.022099	0.054558	0.020043	0
3	0.009761	0.300635	0.017443	0.004748	0.010681	0.028191	0.020967	0
4	0.001000	0.259325	0.034188	0.000000	0.004052	0.006732	0.157735	0
...
9261	0.002240	0.230451	0.278181	0.276533	0.000000	0.000000	0.137091	0
9262	0.000440	0.090072	0.210172	0.114413	0.000000	0.000000	0.117271	0
9263	0.000560	0.240742	0.060096	0.087240	0.000737	0.003647	0.014563	0
9264	0.252190	0.763426	0.000000	0.000220	0.011418	0.031417	0.111715	0

חלק 3:

שאלה 11

```
In [122]: #num 11
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [173]: #num 11
def create_kmeans_classifier(K):
    if ((type(K) == int or float) and (float(K) == int(K)) and (int(K) > 0)):
        return KMEANS(int(K))
    else:
        print("Invalid input!")
```

שאלה 12 + 13

```
In [22]: #num 12 +13

k = range(1, 15)
error = []
n_error = []

def KMEANS(K):
    kmeans = KMeans(n_clusters= K)
    kmeans.fit(df1)
    return kmeans.inertia_

for i in k:
    error.append(create_kmeans_classifier(i))
```

```
In [23]: #num 12 +13

def KMEANS(K):
    kmeans = KMeans(n_clusters= K)
    kmeans.fit(X_normalize)
    return kmeans.inertia_

for i in k:
    n_error.append(create_kmeans_classifier(i))
```

```
In [24]: print(error, "\n" , n_error)
```

שאלה 14

```
In [26]: # num 14
Silhouette = []
n_Silhouette = []

def KMEANS(K):
    kmeans = KMeans(n_clusters= K)
    kmeans.fit(df1)
    return kmeans.cluster_centers_

for i in k:
    Silhouette.append(create_kmeans_classifier(i))
```

```
In [30]: #num 14

def KMEANS(K):
    kmeans = KMeans(n_clusters= K)
    kmeans.fit(X_normalize)
    return kmeans.cluster_centers_

for i in k:
    n_Silhouette.append(create_kmeans_classifier(i))
```

שאלה 15

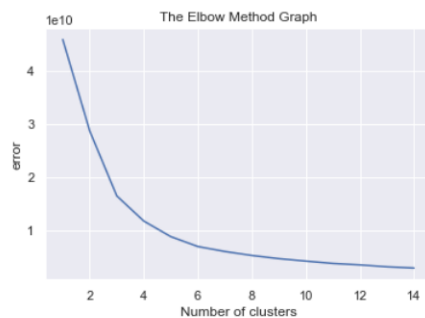
```
In [27]: #num 15
|
| print( "", len(Silhouette), "\n" , len(n_Silhouette), "\n" , len(error), "\n" , len(n_error))
|
| 14
| 14
| 14
| 14
```

חלק 4:

שאלה 16

```
In [66]: plt.title('The Elbow Method Graph')
plt.xlabel('Number of clusters')
plt.ylabel('error')
plt.plot(k, error)
```

Out[66]: [matplotlib.lines.Line2D at 0x241806474c0>]



שאלה 17

```
In [97]: #num 17
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
#Load Data
data = df1
pca = PCA(2)

#Transform the data
df = pca.fit_transform(data)

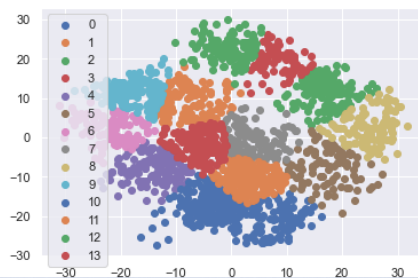
#Import KMeans module
from sklearn.cluster import KMeans

#Initialize the class object
kmeans = KMeans(n_clusters= 14)

#predict the labels of clusters.
label = kmeans.fit_predict(df)

#Getting unique labels
u_labels = np.unique(label)

#plotting the results:
for i in u_labels:
    plt.scatter(df[label == i , 0] , df[label == i , 1] , label = i)
plt.legend()
plt.show()
```



שאלה 18

```
In [98]: #num 18
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
#Load Data
data = X_normalize
pca = PCA(2)

#Transform the data
df = pca.fit_transform(data)

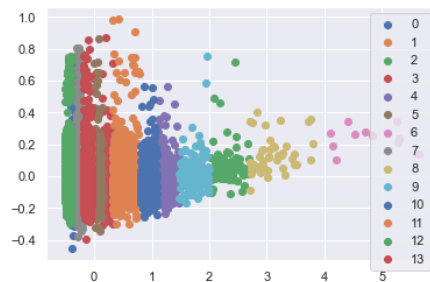
#Import KMeans module
from sklearn.cluster import KMeans

#Initialize the class object
kmeans = KMeans(n_clusters= 14)

#predict the Labels of clusters.
label = kmeans.fit_predict(df)

#Getting unique Labels
u_labels = np.unique(label)

#Plotting the results:
for i in u_labels:
    plt.scatter(df[label == i , 0] , df[label == i , 1] , label = i)
plt.legend()
plt.show()
```



הסבר:

ניתן לראות שכאשר הנתונים מנורמלים אז השוני בין הקבוצות הופך להיות יותר בצורה דו ממדית – ולכן בסכימה של ערכים לכול טווח ערכים ניתן לקטלג אותם יותר בקלות, דבר זה יכול להקל עלינו כאשר הנתונים מאוד קרובים וההשוואה בין הקבוצות לא ברורה לנו חשיבות ערכי הנתונים. כלומר בעזרת נרמול אנחנו יכולים גם להוריד את סיבוכיות החישוב בסדר גודל אחד מ - $o(n^3) \Rightarrow o(n^2)$ וגם לנו יהיה יותר קל להשוות את זה בעיין כאשר מורידים לנו ממד להשוות.

חלק 5 – Excel :

שאלה 19

לכול מדינה נגדיר ווקטור שמורכב ממספרי האחדים וכול אחד "מחזק" את גדול הווקטור ואת הכיוון שלו. לאחר מכן ניתן לראות כול ווקטור כמזה חח"ע למדינה ולמדוד אותה לפי ערכיו.

שאלה 20

[illegible]

שאלה 21

[illegible]

שאלה 22

שאלה 23

הקלד כאן כדי לחפש

שאלה 25 + 24

Excel screenshot showing a spreadsheet with data and formulas. The formula bar shows $=MATCH(F31,F26:F29,0)$.

	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	גידולי סוכר?
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	סוכר וממתקים
19	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	אגודי עץ?
20	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	מוצרים צמחיים?
21	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	שמים צמחיים?
22	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	ירקות?
23	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	מקורות אחרים?
24																	
25																	
26	45751	2.236068	1.732051	2.44949	2.645751	1.732051	1.732051	1.732051	1	1.414214	2.44949	2.236068					מרחק מסנטרואיד 1
27	45751	2.645751	1.732051	2.44949	2.645751	2.236068	2.236068	1.732051	1	2	2.44949	2.236068					מרחק מסנטרואיד 2
28	45751	2.645751	1.732051	2.44949	3	2.236068	2.236068	1.732051	1.732051	2	2.44949	2.236068					מרחק מסנטרואיד 3
29	45751	2.645751	1.732051	2.44949	3	2.236068	2.236068	1.732051	1.732051	2	2.44949	2.236068					מרחק מסנטרואיד 4
30																	
31	45751	2.236068	1.732051	2.44949	2.645751	1.732051	1.732051	1.732051	1	1.414214	2.44949	2.236068					מרחק מינימלי
32	1	1	1	1	1	1	1	1	1	1	1	1					התאמה לסנטרואיד
33																	

שאלה 26

פונקציית המטרה שלנו היא התא של סכום המרחקים $SUM(F31:FS31)$ שאותה אנחנו רוצים להביא למינימום, כך שפשוט נזיז את כול המיקומים של הסנטרואידים. האילוצים שנקבעה הם - שמיקומי המרחקים יהיה קטן שווה ל 1. וגם שמבחינת איטרציות שיהיה קטן ממיליון כדי י שהאקסל לא יתקע לנו.

שאלה 27

שמירה אוטומטית

corona_stats - Excel

כללי

קובץ בית הוספה ציור פרסת עמוד נוסחאות נתונים סקירה תצוגה עזרה

רשימות חפשי מיון וסדר עריכה

תאים

סגנונות

מספר

ישר

גופן

לוח

F35 =SUMSQ(G31:FT31)

	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
1																
2																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																

100% +

הגדרות תצוגה

מוכן

17:37 עבר 24/05/2021 HE

הקלד כאן כדי לחפש

שאלה 28

שמירה אוטומטית

corona_stats - Excel

כללי

קובץ בית הוספה ציור פרסת עמוד נוסחאות נתונים סקירה תצוגה עזרה

רשימות חפשי מיון וסדר עריכה

תאים

סגנונות

מספר

ישר

גופן

לוח

F35 =SUMSQ(G31:FT31)

	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
1																
2																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																

100% +

הגדרות תצוגה

מוכן

17:46 עבר 24/05/2021 HE

הקלד כאן כדי לחפש

שאלה 29

Q15 Q13 Q12 Q11 Q4 Q3 Q2 נתונים טבלאות פריסת עמוד ציר הוספה בית קבץ

מילון סטטיסטיקת קורונה נתונים

100% הגדרות תצוגה

18:02 24/05/2021 ENG

הקלד כאן כדי לחפש

מזכיר

מוצרים	סנטוראיד 1	סנטוראיד 2	סנטוראיד 3	סנטוראיד 4
מהחי?	0.47547	0.027748	0.21867	0.133051
שומן	0.077886	0.028726	0.040421	0.071988
מוצרים	0.025006	0.471554	0.175698	0.798706
מהחי?				
דגנים	0.362727	0.046321	0.164134	0.28314
(ללא	0.204053	0.33922	0.126362	0.121803
בירה)?				
ביצים?	0.058734	0.103644	0.091358	0.740657
דגים				
ופירות ים?				
פירות	0.482406	0.024407	0.011545	0.118093
(ללא יו)?	0.471168	0.002963	0.137195	0.225536
בשר?	0.227758	0.175333	0.257589	0.695532
חלב (ללא				
חמאה)?	0.049739	0.502947	0.048629	0.085795
חלקי				
פנים?				
שמים	0.036287	0.579153	0.044826	0.532827
המפוסקים	0.181931	0.249788	0.089985	0.041472
מקדונות?	0.042104	0.467184	0.04645	0.392071
תבלינים?	0.389935	0.03683	0.02887	0.025804
שורשים	0.050176	0.072538	0.021781	0.083481
עמילניים?				
ממריצים?	0.056709	0.09861	0.022263	0.036832
גידולי				
סוכר?				
סוג				

[illegible]

א - ניתן לראות שבקלאסטר 1 - יש 84 מדינות. בקלאסטר 2 - יש 33 מדינות. בקלאסטר 3 - יש 34 מדינות.
בקלאסטר 1 - יש 19 מדינות

ב- ניתן לראות שבקלאסטר 1 - עיקר השומן מגיע מהחיי ביחד עם ששמינים צמחיים – כמובן שביחס לקבוצות האחרות, כמובן ניתן לראות שהרבה ממדינות אלה הן מדינות דרום אמריקאיות או ברית המועצות לשעבר, מדינות של רוב נמנות כמדינות "מתפתחות = עולם שני" שאינם מתאפיינות במזון מסורתי כמו במדינות עולם שלישי, אך גם לא מאופיינות כמדינות מפותחות. כמובן שניתן לראות גם מדינות מפותחות כמו המדינות הסקנדינביות צרפת ושווייץ שכנראה מאופיינות בצריכת שומן יחסית גבוהה מהחיי.

ניתן לראות שבקלאסטר 2 - עיקר השומן מגיע מאגוזים, תבלינים, שומנים שמגיעים מגידולים, חלקי פנים ומוצרי חלב – כמובן שביחס לקבוצות האחרות. ניתן לראות שכול המדינות באופן מוחלט מוגדרות כמדינות עולם שלישי ולכן מאופיינות ע"י בישול מסורתי, יחסית מגוון וכול זאת לצד עוני קשה. כמובן ניתן לראות שהיה אפשר להסיק בזכות נתוני התמותה והחולי הנמוכים שמזון ושומן מסוג זה יכולים להבטיח עמידות בפני המגפה – אך חשוב לזכור שבמדינות אלה תוחלת החיים היא נמוכה בעד 50% ממדינות המערב (ולכן אחוז זקנים נמוך), לצד מערכת בריאות כושלת וחוסר יכולת לאסוף נתונים בצורה אמינה.

ניתן לראות שבקלאסטר 3 - עיקר השומן ממקורות צמחיים, ומסוכרים ו"ממקורות אחרים" בעוד שאר הערכים יחסית נמוכים – כמובן שביחס לקבוצות האחרות, ניתן לראות שהרבה מדינות מערביות נכללות בקרב קבוצה זאת דבר שמתאפיין באוכל תעשייתי עשיר בסוכר לצד מוצרים צמחיים. כמובן ניתן לראות שאחוז השמנת היתר הינו גבוהה ביותר בקבוצה זאת, לצד תמותה וחולי מקורנה. אבל שוב יש לזכור שלצד נתוני התזונה, תוחלת החיים במדינות אלה היא מהגבוהות בעולם – דבר המשפיע על אחוז הזקנים ועל היכולת לעקוב אחרי המצב הרפואי של האוכלוסייה. ולכן ישפיע גם על הנתונים.

ניתן לראות שבקלאסטר 4 - עיקר השומן מגיע ממוצרים מהים, מוצרי חלב, שמנים מגידולים ותבלינים – כמובן שביחס לקבוצות האחרות. דבר הדומה קצת למדינות קלאסטר 2 שבהם המדינות מתפתחות לצד זאת שמדינות 4 הן בעיקר מדינות מתפתחות בהם רוב האוכלוסיה שוכנת לצד החופים ואגמים לכן מרבית התזונה בהם היא מאכלי ים. בדומה למדינות 2 מדובר במדינות עם מערכות בריאות כושלות לצד תוחלת חיים יחסית נמוכה. וכמובן לצד יכולת לאסוף נתונים בצורה אמינה ולכן יש לבחון את הנתונים במדינות אלה בערבון מוגבל.

חלק 3 שאלה 31

עזרה אוטומטית

קובץ בית הוספה ציור פריסת עמוד נוסחאות נתונים סקירה תצוגה עזרה

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

מספר

<