In [16]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

קריאה של נתונים תתבצע על ידי pandas

In [326…
```
data = pd.read_csv("adult.csv")
data_sampled = data.sample(1000)
```

עיבוד בסיסי של נתונים

In [330…
```
# 5 שורות ראשונות
data.head()
```

Out[330…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	genc
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	M
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	M
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	M
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	M
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fem

In [331…
```
# 5 שורות אחרונות
data.tail()
```

Out[331…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White

In [79]:
```
# מחזיר את מספר השורות והעמודות בנתונים
# במקום ה0 יהיה השורות ובמקום 1 יהיה העמודות
data.shape
```

Out[79]:
```
(48842, 15)
```

In [81]:
```
# מחזיר אינפורמציה של סוג הנתונים שיש לנו - האם זה מספרים או מחרוזת
# int / float - numbers
# object - string
# datetime - תאריך
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
age                48842 non-null int64
workclass          48842 non-null object
fnlwgt             48842 non-null int64
education          48842 non-null object
educational-num    48842 non-null int64
marital-status     48842 non-null object
occupation         48842 non-null object
relationship       48842 non-null object
race               48842 non-null object
gender             48842 non-null object
capital-gain       48842 non-null int64
capital-loss       48842 non-null int64
hours-per-week     48842 non-null int64
native-country     48842 non-null object
income             48842 non-null object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

שינוי סוג עמודה על ידי astype

In [328…
```
print(data['educational-num'].dtype)
data['educational-num'] = data['educational-num'].astype('str')
print(data['educational-num'].dtype)
data['educational-num'] = data['educational-num'].astype('int64')
print(data['educational-num'].dtype)
```

```
int64
object
int64
```

In [82]:
```
# שמועות העמודות
print(data.columns)
# שמות השורות
print(data.index)
```

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'educational-num',
```

```
       'marital-status', 'occupation', 'relationship', 'race', 'gender',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'income'],
      dtype='object')
RangeIndex(start=0, stop=48842, step=1)
```

In [84]:

```
# קריאה של עמודה
data.age
data['age']
```

Out[84]:

```
0        25
1        38
2        28
3        44
4        18
5        34
6        29
7        63
8        24
9        55
10       65
11       36
12       26
13       58
14       48
15       43
16       20
17       43
18       37
19       40
20       34
21       34
22       72
23       25
24       25
25       45
26       22
27       23
28       54
29       32
         ..
48812    30
48813    34
48814    54
48815    37
48816    22
48817    34
48818    30
48819    38
48820    71
48821    45
48822    41
48823    72
48824    45
48825    31
48826    39
48827    37
48828    43
48829    65
48830    43
48831    43
48832    32
48833    43
48834    32
```

```
48835    53
48836    22
48837    27
48838    40
48839    58
48840    22
48841    52
Name: age, Length: 48842, dtype: int64
```

In [291…

```
# קריאה של כמה עמודות
cols = ['age', 'race']
data[cols]
```

Out[291…

	age	race
0	25	Black
1	38	White
2	28	White
3	44	Black
4	18	White
5	34	White
6	29	Black
7	63	White
8	24	White
9	55	White
10	65	White
11	36	White
12	26	White
13	58	White
14	48	White
15	43	White
16	20	White
17	43	White
18	37	White
19	40	Asian-Pac-Islander
20	34	White
21	34	Black
22	72	White
23	25	White
24	25	White
25	45	White
26	22	White
27	23	Black

	age	race
28	54	White
29	32	White
...	...	...
48812	30	Asian-Pac-Islander
48813	34	White
48814	54	Asian-Pac-Islander
48815	37	White
48816	22	Black
48817	34	White
48818	30	Black
48819	38	Black
48820	71	White
48821	45	White
48822	41	Black
48823	72	White
48824	45	White
48825	31	Other
48826	39	White
48827	37	White
48828	43	White
48829	65	White
48830	43	White
48831	43	White
48832	32	Amer-Indian-Eskimo
48833	43	White
48834	32	Asian-Pac-Islander
48835	53	White
48836	22	White
48837	27	White
48838	40	White
48839	58	White
48840	22	White
48841	52	White

48842 rows × 2 columns

In [91]:
```
# Loc
```

```
# צריכים לציין לו את שמות השורות ושמות העמודות
data.loc[[0, 5, 100, 211], ['age', 'gender']]
# 0,5,100,211 כל העמודות והשורות
data.loc[[0, 5, 100, 211], :]
# השורות 0,5,100,211 והעמודות מרייס עד הסוף
data.loc[[0, 5, 100, 211], 'race':]
# iloc
# מחזיר את הערכים של השורות והעמודות רק על ידי מספרים (האינדקס של השורה והעמודה)
data.iloc[1:10, 3:5]
data.iloc[1:20:2, 4:7:2]
```

Out[91]:

	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	Black	Male	0	0	40	United-States	<=50K
5	White	Male	0	0	30	United-States	<=50K
100	White	Female	0	0	18	United-States	<=50K
211	White	Female	0	0	50	United-States	<=50K

In [272…

```
a = data['gender']=='Male'
a.index
```

Out[272…

```
RangeIndex(start=0, stop=48842, step=1)
```

In [273…

```
# סינון של ערכים יתבצע על ידי:
# מתוך הדאטה אילו נתונים נרצה להשאיר
# מה שיש בפנים זה התנאים
print(data.shape)
new_data = data[data['gender']=='Male']
print(new_data.shape)
# אחרי ביצוע הפילטר אנחנו יכולים לקרוא לנתונים ספציפים מתוך מה שנשאר
new_data.loc[:, ['age', 'race']]
```

```
(48842, 16)
(32650, 16)
```

Out[273…

	age	race
0	25	Black
1	38	White
2	28	White
3	44	Black
5	34	White
6	29	Black
7	63	White
9	55	White
10	65	White
11	36	White
13	58	White
14	48	White
15	43	White

	age	race
16	20	White
19	40	Asian-Pac-Islander
20	34	White
23	25	White
24	25	White
25	45	White
26	22	White
27	23	Black
28	54	White
29	32	White
30	46	Black
32	24	White
33	23	White
35	65	White
36	36	White
37	22	White
38	17	White
...	...	...
48787	38	White
48788	50	White
48791	39	White
48793	20	White
48795	40	Black
48796	66	White
48798	36	White
48799	57	White
48800	46	White
48802	33	Black
48803	58	White
48804	30	White
48807	32	White
48808	22	White
48813	34	White
48814	54	Asian-Pac-Islander
48816	22	Black
48818	30	Black

	age	race
48820	71	White
48823	72	White
48828	43	White
48829	65	White
48831	43	White
48832	32	Amer-Indian-Eskimo
48833	43	White
48834	32	Asian-Pac-Islander
48835	53	White
48836	22	White
48838	40	White
48840	22	White

32650 rows × 2 columns

In [280…

```
data.loc[0, 'gender'] = 'male'
```

Out[280…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	genc
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	m
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	M
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	M
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	M
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fem

In [277…

```
data[data.gender.isin(['Male', 'male'])]
```

Out[277…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Blac

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Whit
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Whit
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Blac
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	Whit
6	29	?	227026	HS-grad	9	Never-married	?	Unmarried	Blac
7	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	Whit
9	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	Whit
10	65	Private	184454	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
11	36	Federal-gov	212465	Bachelors	13	Married-civ-spouse	Adm-clerical	Husband	Whit
13	58	?	299831	HS-grad	9	Married-civ-spouse	?	Husband	Whit
14	48	Private	279724	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
15	43	Private	346189	Masters	14	Married-civ-spouse	Exec-managerial	Husband	Whit
16	20	State-gov	444554	Some-college	10	Never-married	Other-service	Own-child	Whit
19	40	Private	85019	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	Asiar Pac Islande
20	34	Private	107914	Bachelors	13	Married-civ-spouse	Tech-support	Husband	Whit
23	25	Private	220931	Bachelors	13	Never-married	Prof-specialty	Not-in-family	Whit
24	25	Private	205947	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Whit

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
25	45	Self-emp-not-inc	432824	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Whit
26	22	Private	236427	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
27	23	Private	134446	HS-grad	9	Separated	Machine-op-inspct	Unmarried	Blac
28	54	Private	99516	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Whit
29	32	Self-emp-not-inc	109282	Some-college	10	Never-married	Prof-specialty	Not-in-family	Whit
30	46	State-gov	106444	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Blac
32	24	Self-emp-not-inc	188274	Bachelors	13	Never-married	Sales	Not-in-family	Whit
33	23	Local-gov	258120	Some-college	10	Married-civ-spouse	Protective-serv	Husband	Whit
35	65	?	191846	HS-grad	9	Married-civ-spouse	?	Husband	Whit
36	36	Local-gov	403681	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Whit
37	22	Private	248446	5th-6th	3	Never-married	Priv-house-serv	Not-in-family	Whit
38	17	Private	269430	10th	6	Never-married	Machine-op-inspct	Not-in-family	Whit
...	...	...	...	...	...	...	...	...	
48787	38	Private	32916	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Whit
48788	50	Private	302372	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Whit
48791	39	Private	107302	HS-grad	9	Married-civ-spouse	Prof-specialty	Husband	Whit
48793	20	Private	270436	HS-grad	9	Never-married	Machine-op-inspct	Own-child	Whit
48795	40	Private	142657	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Blac

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
48796	66	Federal-gov	47358	10th	6	Married-civ-spouse	Craft-repair	Husband	Whit
48798	36	Private	131459	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	Whit
48799	57	Local-gov	110417	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Whit
48800	46	Private	364548	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Whit
48802	33	Private	273243	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Blac
48803	58	Private	147707	11th	7	Married-civ-spouse	Sales	Husband	Whit
48804	30	Private	77266	HS-grad	9	Divorced	Transport-moving	Not-in-family	Whit
48807	32	Private	211349	10th	6	Married-civ-spouse	Transport-moving	Husband	Whit
48808	22	Private	203715	Some-college	10	Never-married	Adm-clerical	Own-child	Whit
48813	34	Private	204461	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	Whit
48814	54	Private	337992	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Asiar Pac Islande
48816	22	Private	325033	12th	8	Never-married	Protective-serv	Own-child	Blac
48818	30	Private	345898	HS-grad	9	Never-married	Craft-repair	Not-in-family	Blac
48820	71	?	287372	Doctorate	16	Married-civ-spouse	?	Husband	Whit
48823	72	?	129912	HS-grad	9	Married-civ-spouse	?	Husband	Whit
48828	43	Private	260761	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
48829	65	Self-emp-not-inc	99359	Prof-school	15	Never-married	Prof-specialty	Not-in-family	Whit

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
48831	43	Self-emp-not-inc	27242	Some-college	10	Married-civ-spouse	Craft-repair	Husband	Whit
48832	32	Private	34066	10th	6	Married-civ-spouse	Handlers-cleaners	Husband	Ame Indiar Eskim
48833	43	Private	84661	Assoc-voc	11	Married-civ-spouse	Sales	Husband	Whit
48834	32	Private	116138	Masters	14	Never-married	Tech-support	Not-in-family	Asiar Pac Islande
48835	53	Private	321865	Masters	14	Married-civ-spouse	Exec-managerial	Husband	Whit
48836	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	Whit
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit

32650 rows × 16 columns

In [283…

```
data[data.relationship.str.contains('-')]
#מציג את כל השורות בעמודת relationship שיש בהן - 
```

Out[283…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Blac
4	18	?	103497	Some-college	10	Never-married	?	Own-child	Whit
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	Whit
12	26	Private	82091	HS-grad	9	Never-married	Adm-clerical	Not-in-family	Whit
16	20	State-gov	444554	Some-college	10	Never-married	Other-service	Own-child	Whit
21	34	Private	238588	Some-college	10	Never-married	Other-service	Own-child	Blac
22	72	?	132015	7th-8th	4	Divorced	?	Not-in-family	Whit

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
23	25	Private	220931	Bachelors	13	Never-married	Prof-specialty	Not-in-family	Whit
26	22	Private	236427	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
29	32	Self-emp-not-inc	109282	Some-college	10	Never-married	Prof-specialty	Not-in-family	Whit
32	24	Self-emp-not-inc	188274	Bachelors	13	Never-married	Sales	Not-in-family	Whit
37	22	Private	248446	5th-6th	3	Never-married	Priv-house-serv	Not-in-family	Whit
38	17	Private	269430	10th	6	Never-married	Machine-op-inspct	Not-in-family	Whit
39	20	Private	257509	HS-grad	9	Never-married	Craft-repair	Own-child	Whit
44	20	State-gov	138371	Some-college	10	Never-married	Farming-fishing	Own-child	Whit
48	52	Private	201062	11th	7	Separated	Priv-house-serv	Not-in-family	Blac
49	56	Self-emp-inc	131916	HS-grad	9	Widowed	Exec-managerial	Not-in-family	Whit
50	18	Private	54440	Some-college	10	Never-married	Other-service	Own-child	Whit
51	39	Private	280215	HS-grad	9	Divorced	Handlers-cleaners	Own-child	Blac
52	21	Private	214399	Some-college	10	Never-married	Other-service	Own-child	Whit
53	22	Private	54164	HS-grad	9	Never-married	Other-service	Not-in-family	Whit
54	38	Private	219446	9th	5	Married-spouse-absent	Exec-managerial	Not-in-family	Whit
55	21	Private	110677	Some-college	10	Never-married	Adm-clerical	Own-child	Whit
60	30	Private	101135	Bachelors	13	Never-married	Exec-managerial	Not-in-family	Whit
61	39	Private	118429	Some-college	10	Divorced	Sales	Not-in-family	Whit
62	26	Private	31208	Masters	14	Never-married	Exec-managerial	Not-in-family	Whit
63	33	Private	281384	HS-grad	9	Never-married	Machine-op-inspct	Own-child	Whit
64	47	Local-gov	171807	HS-grad	9	Divorced	Adm-clerical	Not-in-family	Whit
65	41	Private	109912	Bachelors	13	Never-married	Other-service	Not-in-family	Whit

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
67	19	Private	105460	Some-college	10	Never-married	Other-service	Own-child	Whit
...	...	...	...	...	...	...	...	...	
48769	32	Private	164190	Some-college	10	Never-married	Exec-managerial	Own-child	Whit
48771	60	?	134152	9th	5	Divorced	?	Not-in-family	Blac
48773	42	Self-emp-not-inc	217597	HS-grad	9	Divorced	Sales	Own-child	Whit
48775	82	?	403910	HS-grad	9	Never-married	?	Not-in-family	Whit
48776	26	Private	179010	Some-college	10	Never-married	Craft-repair	Not-in-family	Whit
48777	18	Private	436163	11th	7	Never-married	Prof-specialty	Own-child	Whit
48778	34	Private	321709	HS-grad	9	Never-married	Other-service	Not-in-family	Whit
48780	25	Private	403788	HS-grad	9	Never-married	Craft-repair	Other-relative	Blac
48785	50	Private	208630	Masters	14	Divorced	Sales	Not-in-family	Whit
48786	33	Private	182401	10th	6	Never-married	Adm-clerical	Not-in-family	Blac
48789	45	Private	155093	10th	6	Divorced	Other-service	Not-in-family	Blac
48790	32	Private	192965	HS-grad	9	Separated	Sales	Not-in-family	Whit
48792	25	Local-gov	514716	Bachelors	13	Never-married	Adm-clerical	Own-child	Blac
48793	20	Private	270436	HS-grad	9	Never-married	Machine-op-inspct	Own-child	Whit
48804	30	Private	77266	HS-grad	9	Divorced	Transport-moving	Not-in-family	Whit
48805	26	Private	191648	Assoc-acdm	12	Never-married	Machine-op-inspct	Other-relative	Whit
48808	22	Private	203715	Some-college	10	Never-married	Adm-clerical	Own-child	Whit
48812	30	?	33811	Bachelors	13	Never-married	?	Not-in-family	Asiar Pac Islande
48816	22	Private	325033	12th	8	Never-married	Protective-serv	Own-child	Blac
48817	34	Private	160216	Bachelors	13	Never-married	Exec-managerial	Not-in-family	Whit

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
48818	30	Private	345898	HS-grad	9	Never-married	Craft-repair	Not-in-family	Blac
48821	45	State-gov	252208	HS-grad	9	Separated	Adm-clerical	Own-child	Whit
48822	41	?	202822	HS-grad	9	Separated	?	Not-in-family	Blac
48825	31	Private	199655	Masters	14	Divorced	Other-service	Not-in-family	Othe
48827	37	Private	198216	Assoc-acdm	12	Divorced	Tech-support	Not-in-family	Whit
48829	65	Self-emp-not-inc	99359	Prof-school	15	Never-married	Prof-specialty	Not-in-family	Whit
48830	43	State-gov	255835	Some-college	10	Divorced	Adm-clerical	Other-relative	Whit
48834	32	Private	116138	Masters	14	Never-married	Tech-support	Not-in-family	Asiar Pac Islande
48836	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	Whit
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit

21670 rows × 16 columns

In [303…

```
data[data.gender.str.startswith('Ma')].shape[0]
```

Out[303…

```
32650
```

In [98]:

```
# איך יוצרים תנאי וגם על הנתונים
data[(data['gender'] == 'Male') & (data['age'] > 30)]
# איך יוצרים תנאי או
data[(data['gender'] == 'Male') | (data['age'] > 30)]
```

Out[98]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
7	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White
9	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White
10	65	Private	184454	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
11	36	Federal-gov	212465	Bachelors	13	Married-civ-spouse	Adm-clerical	Husband	White
13	58	?	299831	HS-grad	9	Married-civ-spouse	?	Husband	White
14	48	Private	279724	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
15	43	Private	346189	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White
19	40	Private	85019	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander
20	34	Private	107914	Bachelors	13	Married-civ-spouse	Tech-support	Husband	White
25	45	Self-emp-not-inc	432824	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White
28	54	Private	99516	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White
29	32	Self-emp-not-inc	109282	Some-college	10	Never-married	Prof-specialty	Not-in-family	White
30	46	State-gov	106444	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black
35	65	?	191846	HS-grad	9	Married-civ-spouse	?	Husband	White
36	36	Local-gov	403681	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
40	65	Private	136384	Masters	14	Married-civ-spouse	Prof-specialty	Husband	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
41	44	Self-emp-inc	120277	Assoc-voc	11	Married-civ-spouse	Sales	Husband	White
42	36	Private	465326	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
46	39	Private	290208	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White
47	54	Private	186272	Some-college	10	Married-civ-spouse	Transport-moving	Husband	White
51	39	Private	280215	HS-grad	9	Divorced	Handlers-cleaners	Own-child	Black
54	38	Private	219446	9th	5	Married-spouse-absent	Exec-managerial	Not-in-family	White
56	63	Private	145985	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White
57	34	Local-gov	382078	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
58	42	Self-emp-inc	170721	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White
59	33	Private	269705	HS-grad	9	Married-civ-spouse	Handlers-cleaners	Husband	White
61	39	Private	118429	Some-college	10	Divorced	Sales	Not-in-family	White
...	...	...	...	...	...	...	...	...	..
48775	82	?	403910	HS-grad	9	Never-married	?	Not-in-family	White
48779	57	Private	153918	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White
48781	34	Private	60567	11th	7	Divorced	Transport-moving	Unmarried	White
48782	71	Private	138145	9th	5	Married-civ-spouse	Other-service	Husband	White
48783	35	Local-gov	79649	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
48784	47	Private	312088	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
48786	33	Private	182401	10th	6	Never-married	Adm-clerical	Not-in-family	Black
48787	38	Private	32916	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	White
48788	50	Private	302372	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
48791	39	Private	107302	HS-grad	9	Married-civ-spouse	Prof-specialty	Husband	White
48795	40	Private	142657	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Black
48796	66	Federal-gov	47358	10th	6	Married-civ-spouse	Craft-repair	Husband	White
48798	36	Private	131459	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White
48799	57	Local-gov	110417	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White
48800	46	Private	364548	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	White
48802	33	Private	273243	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Black
48803	58	Private	147707	11th	7	Married-civ-spouse	Sales	Husband	White
48807	32	Private	211349	10th	6	Married-civ-spouse	Transport-moving	Husband	White
48813	34	Private	204461	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	White
48814	54	Private	337992	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Asian-Pac-Islander
48820	71	?	287372	Doctorate	16	Married-civ-spouse	?	Husband	White
48823	72	?	129912	HS-grad	9	Married-civ-spouse	?	Husband	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
48828	43	Private	260761	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
48829	65	Self-emp-not-inc	99359	Prof-school	15	Never-married	Prof-specialty	Not-in-family	White
48831	43	Self-emp-not-inc	27242	Some-college	10	Married-civ-spouse	Craft-repair	Husband	White
48832	32	Private	34066	10th	6	Married-civ-spouse	Handlers-cleaners	Husband	Amer-Indian-Eskimo
48833	43	Private	84661	Assoc-voc	11	Married-civ-spouse	Sales	Husband	White
48834	32	Private	116138	Masters	14	Never-married	Tech-support	Not-in-family	Asian-Pac-Islander
48835	53	Private	321865	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White

23265 rows × 15 columns

על משתנים שהם מספריים נעשה עיבוד של ממוצע, חציון והשכיח ביותר

In [ ]:

```
# mean, median, mode
print(data['age'].mean())
print(data['age'].median())
print(data['age'].mode())
# more acions: .max(), min()
print(data['age'].max())
print(data['age'].min())
print(data['age'].count())
```

מציאת הערכים היחודיים שיש לנו בעמודה בלבד:

In [292…

```
data.gender.unique()
```

Out[292…

```
array(['male', 'Male', 'Female'], dtype=object)
```

שינוי ערך שיש לנו בנתונים

In [294…

```
data = data.replace({'male': 'Male'})
print(data.gender.unique())
```

```
['Male' 'Female']
```

כמה יש בכל קבוצה

In [239…
```
data_sampled.gender.value_counts()
```

Out[239…
```
Male      651
Female    349
Name: gender, dtype: int64
```

שאלה פתוחה - איך נעשה את אחוז הגברים ואחוז הנשים בדאטה שלנו ?

In [ ]:
```
????????
```

ממוצע הרמוני - סוג של ממוצע משוקלל, כאשר אין לנו יכולת לעשות ממוצע רגיל (כמו נסיעה לשני הכיוונים במהירות שונה) נעשה ממוצא הרמוני נעשה:

- (כמות המשתנים / (1/ערך ראשון + 1/ ערך שני + 1/ ערך שלישי

In [9]:
```
print("not true ", (80 + 100) / 2)
print("true", 2 / (1/80 + 1/100))
```

```
 not true  90.0
true 88.88888888888889
```

ערכים חסרים:

In [131…
```
# סתם קריאת נתונים לטובת התרגיל
url = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vacci
vacc_df = pd.read_csv(url)
```

In [104…
```
vacc_df.head()
```

Out[104…

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	total_
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	

In [107…
```
# איך יודעים כמה ערכים חסרים יש בכל עמודה ?
vacc_df.isnull().sum()
# כמה ערכים חסרים יש בעמודה ספציפית?
print(vacc_df.daily_vaccinations.isnull().sum())
print(vacc_df['daily_vaccinations'].isnull().sum())
print(vacc_df.loc[:, 'daily_vaccinations'].isnull().sum())
```

```
321
321
```

```
321
```

מה עושים עם הערכים החסרים ?

- לזרוק אותם

In [118…

```
# נשתמש ב dropna
vacc_df.dropna()
# חשוב לשים לב שהוא לא שומר את הערכים רק אם נעשה שמירה לתוך הקובץ שלנו
# vacc_df = vacc_df.dropna()
vacc_df.isnull().sum()
```

Out[118…

```
location                               0
iso_code                               0
date                                   0
total_vaccinations                     0
people_vaccinated                      0
people_fully_vaccinated                0
total_boosters                         0
daily_vaccinations_raw                 0
daily_vaccinations                     0
total_vaccinations_per_hundred         0
people_vaccinated_per_hundred          0
people_fully_vaccinated_per_hundred    0
total_boosters_per_hundred             0
daily_vaccinations_per_million         0
dtype: int64
```

לזרוק רק עמודות ספציפיות

In [132…

```
print(vacc_df.isnull().sum())
# החלטנו ששורה ספציפית לא רלוונטית מכיוון שיש לה המון ערכים חסרים
# נרצה לזרוק אותה ספציפית
print('---------------')
vacc_df = vacc_df.drop(columns=['total_boosters_per_hundred'])
print(vacc_df.isnull().sum())
```

```
location                                   0
iso_code                                   0
date                                       0
total_vaccinations                     21202
people_vaccinated                      22351
people_fully_vaccinated                25499
total_boosters                         46937
daily_vaccinations_raw                 25774
daily_vaccinations                       321
total_vaccinations_per_hundred         21202
people_vaccinated_per_hundred          22351
people_fully_vaccinated_per_hundred    25499
total_boosters_per_hundred             46937
daily_vaccinations_per_million           321
dtype: int64
---------------
location                                   0
iso_code                                   0
date                                       0
total_vaccinations                     21202
people_vaccinated                      22351
people_fully_vaccinated                25499
total_boosters                         46937
daily_vaccinations_raw                 25774
daily_vaccinations                       321
total_vaccinations_per_hundred         21202
people_vaccinated_per_hundred          22351
```

```
people_fully_vaccinated_per_hundred    25499
daily_vaccinations_per_million           321
dtype: int64
```

מה עם לא נרצה לזרוק ? אלא נרצה לשמור את העמודה אבל למלא את הערכים החסרים?

In [133…

```
# כאשר נרצה להכניס במקום הערכים החסרים ערך ספציפי נשתמש ב:
# fillna()
# או שניתן ערך ספציפי
print(vacc_df.head())
print('------')
vacc_df['total_boosters'] = vacc_df.total_boosters.fillna('eliya')
vacc_df.head()
```

```
      location iso_code        date  total_vaccinations  people_vaccinated  \
0  Afghanistan      AFG  2021-02-22                 0.0                0.0
1  Afghanistan      AFG  2021-02-23                 NaN                NaN
2  Afghanistan      AFG  2021-02-24                 NaN                NaN
3  Afghanistan      AFG  2021-02-25                 NaN                NaN
4  Afghanistan      AFG  2021-02-26                 NaN                NaN

   people_fully_vaccinated  total_boosters  daily_vaccinations_raw  \
0                      NaN             NaN                     NaN
1                      NaN             NaN                     NaN
2                      NaN             NaN                     NaN
3                      NaN             NaN                     NaN
4                      NaN             NaN                     NaN

   daily_vaccinations  total_vaccinations_per_hundred  \
0                 NaN                             0.0
1              1367.0                             NaN
2              1367.0                             NaN
3              1367.0                             NaN
4              1367.0                             NaN

   people_vaccinated_per_hundred  people_fully_vaccinated_per_hundred  \
0                            0.0                                  NaN
1                            NaN                                  NaN
2                            NaN                                  NaN
3                            NaN                                  NaN
4                            NaN                                  NaN

   daily_vaccinations_per_million
0                             NaN
1                            34.0
2                            34.0
3                            34.0
4                            34.0
------
```

Out[133…

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	total_
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	

In [140…

```
# כאשר נרצה להכניס במקום הערכים החסרים ערך ספציפי נשתמש ב:
# forward fill
print(vacc_df.loc[52:62, 'total_vaccinations'])
print(vacc_df[['total_vaccinations']].fillna(method='ffill')[52:62])
print('-------------------')
# backward fill
print(vacc_df.loc[52:62, 'total_vaccinations'])
vacc_df[['total_vaccinations']].fillna(method='bfill')[52:62]
```

```
52         NaN
53         NaN
54         NaN
55         NaN
56         NaN
57         NaN
58         NaN
59    240000.0
60         NaN
61         NaN
62         NaN
Name: total_vaccinations, dtype: float64
    total_vaccinations
52            120000.0
53            120000.0
54            120000.0
55            120000.0
56            120000.0
57            120000.0
58            120000.0
59            240000.0
60            240000.0
61            240000.0
-------------------
52         NaN
53         NaN
54         NaN
55         NaN
56         NaN
57         NaN
58         NaN
59    240000.0
60         NaN
61         NaN
62         NaN
Name: total_vaccinations, dtype: float64
```

Out[140…

	total_vaccinations
52	240000.0
53	240000.0
54	240000.0
55	240000.0
56	240000.0
57	240000.0
58	240000.0
59	240000.0

	total_vaccinations
60	504502.0
61	504502.0

נניח נרצה לתת לערכים החסרים את הערך ההמוצע של כל עמודה?

In [144…

```
print(vacc_df.isnull().sum())
avg = vacc_df.daily_vaccinations_per_million.mean()
vacc_df['daily_vaccinations_per_million'] = vacc_df.daily_vaccinations_per_million.f
print('--------------------')
print(vacc_df.isnull().sum())
```

```
location                                 0
iso_code                                 0
date                                     0
total_vaccinations                   21202
people_vaccinated                    22351
people_fully_vaccinated              25499
total_boosters                           0
daily_vaccinations_raw               25774
daily_vaccinations                     321
total_vaccinations_per_hundred       21202
people_vaccinated_per_hundred        22351
people_fully_vaccinated_per_hundred  25499
daily_vaccinations_per_million         321
dtype: int64
--------------------
location                                 0
iso_code                                 0
date                                     0
total_vaccinations                   21202
people_vaccinated                    22351
people_fully_vaccinated              25499
total_boosters                           0
daily_vaccinations_raw               25774
daily_vaccinations                     321
total_vaccinations_per_hundred       21202
people_vaccinated_per_hundred        22351
people_fully_vaccinated_per_hundred  25499
daily_vaccinations_per_million           0
dtype: int64
```

שיטה נוספת זה interpolate

In [284…

```
s = pd.Series([0, 2, np.nan, 6])
s.interpolate()
```

Out[284…

```
0    0.0
1    2.0
2    4.0
3    6.0
dtype: float64
```

In [162…

```
vacc_df['newTotal2'] = vacc_df['total_vaccinations'].interpolate(method='linear')
```

אם נרצה לעשות פעולה על כל קבוצה מה שנעשה זה groupby

In [146…

```
vacc_df.head()
```

Out[146...

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	total_
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	

In [147...

```
vacc_df.groupby(['location']).mean()
```

Out[147...

location	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
Afghanistan	7.887411e+05	4.724756e+05	1.646886e+05	3.437000e+03
Africa	3.932311e+07	2.657377e+07	1.446285e+07	5.702576e+05
Albania	7.478515e+05	5.568703e+05	4.456750e+05	8.621938e+03
Algeria	3.701797e+06	2.385225e+06	2.633145e+06	3.000000e+01
Andorra	3.110790e+04	2.297300e+04	1.565547e+04	2.401000e+03
Angola	1.099336e+06	7.172499e+05	5.383940e+05	NaN
Anguilla	1.215264e+04	7.349320e+03	7.063706e+03	1.421000e+03
Antigua and Barbuda	5.605031e+04	3.485490e+04	2.672465e+04	9.007778e+02
Argentina	1.635457e+07	1.229000e+07	4.470206e+06	2.074071e+05
Armenia	1.415374e+05	1.002471e+05	6.696131e+04	NaN
Aruba	1.169919e+05	6.538031e+04	5.161155e+04	8.614522e+02
Asia	1.213499e+09	6.256214e+08	2.907329e+08	1.463987e+07
Australia	7.340409e+06	6.880172e+06	4.048052e+06	1.167410e+05
Austria	4.745837e+06	2.835744e+06	2.148141e+06	3.994108e+04
Azerbaijan	3.445169e+06	2.089525e+06	1.454236e+06	3.959243e+04
Bahamas	7.809883e+04	4.617285e+04	4.047315e+04	1.243500e+03
Bahrain	1.581112e+06	7.088232e+05	7.690519e+05	9.527768e+03
Bangladesh	1.249218e+07	8.428665e+06	5.430002e+06	3.045964e+05
Barbados	1.289348e+05	8.128275e+04	7.059569e+04	1.126495e+03
Belarus	1.420250e+06	8.267661e+05	6.594271e+05	NaN
Belgium	6.801974e+06	4.075801e+06	2.865722e+06	6.179701e+04
Belize	1.055588e+05	7.923449e+04	4.499058e+04	2.091863e+03

location	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
Benin	7.983500e+04	6.274891e+04	1.210571e+04	NaN
Bermuda	5.213093e+04	2.798769e+04	2.677568e+04	NaN
Bhutan	5.829263e+05	4.667657e+05	3.470826e+05	1.249323e+04
Bolivia	1.958787e+06	1.395218e+06	7.560256e+05	2.461083e+04
Bonaire Sint Eustatius and Saba	2.161800e+04	1.241750e+04	9.200500e+03	NaN
Bosnia and Herzegovina	4.593401e+05	2.979034e+05	1.721991e+05	4.693100e+04
Botswana	2.285451e+05	1.560402e+05	1.292479e+05	1.043500e+04
Brazil	7.239132e+07	5.135426e+07	2.417944e+07	8.927971e+05
...	...	...	...	...
Tajikistan	1.014541e+06	7.888647e+05	2.590998e+05	6.237250e+04
Tanzania	2.447422e+05	2.447422e+05	2.447422e+05	NaN
Thailand	1.306417e+07	9.444147e+06	3.580514e+06	2.697758e+05
Timor	3.076175e+05	2.260851e+05	9.512111e+04	NaN
Togo	3.316892e+05	2.337539e+05	1.591449e+05	NaN
Tokelau	6.403333e+02	6.403333e+02	NaN	NaN
Tonga	4.378488e+04	2.917175e+04	1.948417e+04	NaN
Trinidad and Tobago	4.019192e+05	2.519386e+05	1.742732e+05	6.372312e+03
Tunisia	2.088389e+06	1.453170e+06	8.376521e+05	3.496676e+04
Turkey	3.888745e+07	2.508739e+07	1.644541e+07	4.247847e+05
Turkmenistan	4.199300e+04	3.224000e+04	9.753000e+03	NaN
Turks and Caicos Islands	3.827123e+04	1.868924e+04	1.869175e+04	NaN
Tuvalu	6.569000e+03	4.656000e+03	3.826000e+03	NaN
Uganda	5.230516e+05	2.706125e+05	1.722655e+05	1.091026e+04
Ukraine	3.440860e+06	2.213320e+06	1.357113e+06	6.050814e+04
United Arab Emirates	1.085307e+07	7.954998e+06	6.978028e+06	7.738947e+04
United Kingdom	5.349884e+07	3.340148e+07	2.041261e+07	3.563887e+05
United States	2.220950e+08	1.287691e+08	1.051156e+08	1.509483e+06
Upper middle income	1.005659e+09	4.617226e+08	2.535632e+08	1.144413e+07
Uruguay	3.118691e+06	1.738985e+06	1.510626e+06	2.946158e+04

location	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw
Uzbekistan	8.224210e+06	5.530520e+06	1.094206e+06	2.537312e+05
Vanuatu	2.159429e+04	1.852614e+04	7.159000e+03	NaN
Venezuela	2.388255e+06	1.542452e+06	1.361631e+06	NaN
Vietnam	7.017551e+06	5.911273e+06	1.372715e+06	2.094352e+05
Wales	2.735836e+06	1.684231e+06	1.068231e+06	1.788863e+04
Wallis and Futuna	7.540680e+03	4.331000e+03	4.012100e+03	NaN
World	1.869933e+09	1.037883e+09	6.005203e+08	2.045640e+07
Yemen	2.228054e+05	2.191246e+05	1.226933e+04	NaN
Zambia	2.743716e+05	1.808307e+05	1.096718e+05	6.351554e+03
Zimbabwe	1.507971e+06	9.584699e+05	6.460799e+05	2.574175e+04

234 rows × 9 columns

In [148…

```
# דוגמא נוספת:
data = pd.read_csv("adult.csv")
data.head()
```

Out[148…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	genc
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	M
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	M
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	M
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	M
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fem

In [163…

```
# נניח נרצה את הגיל ההמוצע של כל הגברים וכל הנשים בנפרד
print(data.groupby(['gender'])['age'].mean())
# נרצה ממוצע לגיל לפי צבע ומגר בנפרד
data.groupby(['gender', 'race'])['age'].mean()
```

```
gender
Female    36.927989
Male      39.494395
Name: age, dtype: float64
```

Out[163…

```
gender  race
Female  Amer-Indian-Eskimo    36.237838
        Asian-Pac-Islander    35.657640
        Black                 37.905979
        Other                 31.212903
        White                 36.882935
Male    Amer-Indian-Eskimo    36.989474
        Asian-Pac-Islander    38.994012
        Black                 37.922592
        Other                 35.167331
        White                 39.704507
Name: age, dtype: float64
```

In [179…

```
a = data.groupby(['gender', 'race'])['age'].mean()#.reset_index()
# בשביל לאפס את האינדקסים נשתמש ב
# reset index
# בעצם לוקח את הכל כעמודות והאינקסים יהיו בסדר עולה
a = a.reset_index()
a[a.age > 35]
```

Out[179…

	gender	race	age
0	Female	Amer-Indian-Eskimo	36.237838
1	Female	Asian-Pac-Islander	35.657640
2	Female	Black	37.905979
4	Female	White	36.882935
5	Male	Amer-Indian-Eskimo	36.989474
6	Male	Asian-Pac-Islander	38.994012
7	Male	Black	37.922592
8	Male	Other	35.167331
9	Male	White	39.704507

עשיית פעולה על עמודה ספציפית - לדוגמא אם נרצה מכל עמודה של שם מסויים לקחת את apply התו השלישי

In [156…

```
data['aviya'] = data['native-country'].apply(lambda x: x[0])
data.sample(10)
```

Out[156…

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
41129	26	Self-emp-not-inc	177858	Bachelors	13	Divorced	Exec-managerial	Not-in-family	White
48538	51	Self-emp-inc	213296	HS-grad	9	Married-civ-spouse	Other-service	Husband	White
8740	30	Self-emp-inc	127651	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
42146	37	Private	295127	Some-college	10	Divorced	Other-service	Not-in-family	White

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
1262	28	Private	184806	Masters	14	Never-married	Exec-managerial	Not-in-family	White
19601	46	Private	180695	Some-college	10	Never-married	Adm-clerical	Own-child	White
7860	30	Private	378009	HS-grad	9	Never-married	Machine-op-inspct	Own-child	White
37418	34	Private	213887	Some-college	10	Married-civ-spouse	Craft-repair	Husband	White
41845	19	Private	64112	12th	8	Never-married	Adm-clerical	Not-in-family	White
39189	30	Private	118056	Some-college	10	Married-spouse-absent	Exec-managerial	Unmarried	White

עבודה עם תאריכים

In [189…

```
# להפוך את העמודה הרלוונטית להיות מסוג תאריך
vacc_df['date'] = pd.to_datetime(vacc_df['date'])
#נוכל לעשות פעולות שונות על העמודה הרלוונטית
# על ידי שימוש ב dt
vacc_df['month'] = vacc_df['date'].dt.month
vacc_df['day'] = vacc_df['date'].dt.day
```

Out[189…

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	total_
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	

דוגמא להבאת כל הנתונים של חודש פבואר דוגמא נוספת של הבאת כל הנתונים של חודש פבואר משנת 2021

In [ ]:

```
vacc_df[(vacc_df['date'].dt.month == 2)]
```

In [310…

```
vacc_df[(vacc_df['date'].dt.month == 3) & (vacc_df['date'].dt.year == 2021)]
```

Out[310…

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	t

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	t
7	Afghanistan	AFG	2021-03-01	NaN	NaN	NaN	
8	Afghanistan	AFG	2021-03-02	NaN	NaN	NaN	
9	Afghanistan	AFG	2021-03-03	NaN	NaN	NaN	
10	Afghanistan	AFG	2021-03-04	NaN	NaN	NaN	
11	Afghanistan	AFG	2021-03-05	NaN	NaN	NaN	
12	Afghanistan	AFG	2021-03-06	NaN	NaN	NaN	
13	Afghanistan	AFG	2021-03-07	NaN	NaN	NaN	
14	Afghanistan	AFG	2021-03-08	NaN	NaN	NaN	
15	Afghanistan	AFG	2021-03-09	NaN	NaN	NaN	
16	Afghanistan	AFG	2021-03-10	NaN	NaN	NaN	
17	Afghanistan	AFG	2021-03-11	NaN	NaN	NaN	
18	Afghanistan	AFG	2021-03-12	NaN	NaN	NaN	
19	Afghanistan	AFG	2021-03-13	NaN	NaN	NaN	
20	Afghanistan	AFG	2021-03-14	NaN	NaN	NaN	
21	Afghanistan	AFG	2021-03-15	NaN	NaN	NaN	
22	Afghanistan	AFG	2021-03-16	54000.0	54000.0	NaN	
23	Afghanistan	AFG	2021-03-17	NaN	NaN	NaN	
24	Afghanistan	AFG	2021-03-18	NaN	NaN	NaN	
25	Afghanistan	AFG	2021-03-19	NaN	NaN	NaN	
26	Afghanistan	AFG	2021-03-20	NaN	NaN	NaN	
27	Afghanistan	AFG	2021-03-21	NaN	NaN	NaN	
28	Afghanistan	AFG	2021-03-22	NaN	NaN	NaN	
29	Afghanistan	AFG	2021-03-23	NaN	NaN	NaN	

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	t
30	Afghanistan	AFG	2021-03-24	NaN	NaN	NaN	
31	Afghanistan	AFG	2021-03-25	NaN	NaN	NaN	
32	Afghanistan	AFG	2021-03-26	NaN	NaN	NaN	
33	Afghanistan	AFG	2021-03-27	NaN	NaN	NaN	
34	Afghanistan	AFG	2021-03-28	NaN	NaN	NaN	
35	Afghanistan	AFG	2021-03-29	NaN	NaN	NaN	
36	Afghanistan	AFG	2021-03-30	NaN	NaN	NaN	
...	...	...	...	...	...	...	
49531	Zimbabwe	ZWE	2021-03-02	25334.0	25334.0	NaN	
49532	Zimbabwe	ZWE	2021-03-03	28227.0	28227.0	NaN	
49533	Zimbabwe	ZWE	2021-03-04	30915.0	30915.0	NaN	
49534	Zimbabwe	ZWE	2021-03-05	31472.0	31472.0	NaN	
49535	Zimbabwe	ZWE	2021-03-06	32251.0	32251.0	NaN	
49536	Zimbabwe	ZWE	2021-03-07	32786.0	32786.0	NaN	
49537	Zimbabwe	ZWE	2021-03-08	36064.0	36064.0	NaN	
49538	Zimbabwe	ZWE	2021-03-09	36307.0	36307.0	NaN	
49539	Zimbabwe	ZWE	2021-03-10	36447.0	36447.0	NaN	
49540	Zimbabwe	ZWE	2021-03-11	36565.0	36565.0	NaN	
49541	Zimbabwe	ZWE	2021-03-12	36829.0	36829.0	NaN	
49542	Zimbabwe	ZWE	2021-03-13	36905.0	36905.0	NaN	
49543	Zimbabwe	ZWE	2021-03-14	36905.0	36905.0	NaN	
49544	Zimbabwe	ZWE	2021-03-15	38206.0	38206.0	NaN	
49545	Zimbabwe	ZWE	2021-03-16	40096.0	40096.0	NaN	

	location	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	t
49546	Zimbabwe	ZWE	2021-03-17	40153.0	40153.0	NaN	
49547	Zimbabwe	ZWE	2021-03-18	40548.0	40548.0	NaN	
49548	Zimbabwe	ZWE	2021-03-19	42233.0	42233.0	NaN	
49549	Zimbabwe	ZWE	2021-03-20	42756.0	42756.0	NaN	
49550	Zimbabwe	ZWE	2021-03-21	43275.0	43275.0	NaN	
49551	Zimbabwe	ZWE	2021-03-22	44120.0	43840.0	280.0	
49552	Zimbabwe	ZWE	2021-03-23	45743.0	44681.0	1062.0	
49553	Zimbabwe	ZWE	2021-03-24	52439.0	49950.0	2489.0	
49554	Zimbabwe	ZWE	2021-03-25	59533.0	55438.0	4095.0	
49555	Zimbabwe	ZWE	2021-03-26	68208.0	61639.0	6569.0	
49556	Zimbabwe	ZWE	2021-03-27	73977.0	66012.0	7965.0	
49557	Zimbabwe	ZWE	2021-03-28	79685.0	69057.0	10628.0	
49558	Zimbabwe	ZWE	2021-03-29	82156.0	70297.0	11859.0	
49559	Zimbabwe	ZWE	2021-03-30	86412.0	73490.0	12922.0	
49560	Zimbabwe	ZWE	2021-03-31	92426.0	77541.0	14885.0	

5316 rows × 16 columns

In [191…

```
vacc_df.groupby(['location', 'month']).total_vaccinations.mean()
```

Out[191…

```
location           month
Afghanistan        2        4.100000e+03
                   3        5.400000e+04
                   4        1.800000e+05
                   5        5.682665e+05
                   6        7.211901e+05
                   7        1.016953e+06
                   8        1.590469e+06
                   9        3.133227e+06
Africa             1        3.118594e+04
                   2        1.820668e+06
                   3        7.409631e+06
                   4        1.505503e+07
                   5        2.467015e+07
```

```
                    6        4.119211e+07
                    7        5.663887e+07
                    8        8.309745e+07
                    9        1.222992e+08
Albania             1        3.329091e+02
                    2        3.923857e+03
                    3        5.603243e+04
                    4        3.101595e+05
                    5        6.499112e+05
                    6        8.592770e+05
                    7        1.057060e+06
                    8        1.339251e+06
                    9        1.594955e+06
Algeria             1        1.500000e+01
                    2        7.500000e+04
                    3                 NaN
                    4                 NaN
                                 ...
Wallis and Futuna   9        1.006650e+04
World               1        4.563287e+07
                    2        1.749979e+08
                    3        4.079767e+08
                    4        8.711842e+08
                    5        1.507635e+09
                    6        2.519170e+09
                    7        3.624229e+09
                    8        4.768158e+09
                    9        5.720999e+09
                    12       2.097080e+06
Yemen               5        6.131250e+04
                    6        2.347214e+05
                    7        3.044440e+05
                    8                 NaN
                    9        3.229340e+05
Zambia              4        1.118373e+04
                    5        1.054705e+05
                    6        1.504658e+05
                    7        2.667901e+05
                    8        5.154320e+05
                    9        6.232967e+05
Zimbabwe            2        8.833875e+03
                    3        4.615116e+04
                    4        2.712150e+05
                    5        7.755129e+05
                    6        1.140177e+06
                    7        1.793201e+06
                    8        3.425626e+06
                    9        4.707671e+06
Name: total_vaccinations, Length: 1841, dtype: float64
```

Pivot table

בעצם נותן לנו את האפשרות לעשות חישובים לפי חיתוכים מסויימים (דומה לקבוצות שעשינו)

In [237…

```
# data_sampled.groupby(['gender', 'race'])['age'].mean()
data_sampled.pivot_table(index='gender', columns='race', values='age', aggfunc='mean
```

Out[237…

race	Amer-Indian-Eskimo	Asian-Pac-Islander	Black	Other	White
gender					
Female	43.666667	37.20	35.890625	28.500000	37.597070
Male	39.090909	32.35	37.740000	44.333333	40.201058

גרפים

לפני שנעבור על גרפים, נבין שעל כל גרף אפשר לעשות את הדברים הבאים:

* לא חובה *

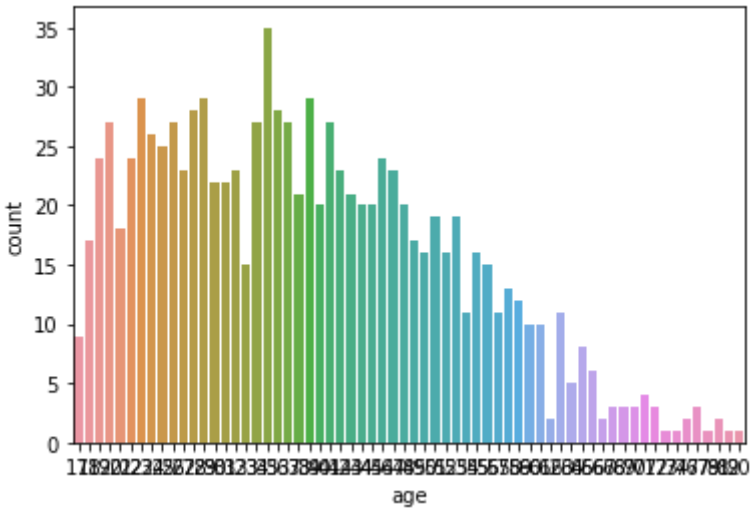
- ax, fig = plt.subplots(rows=2, cols=3, figsize=(10,15)) - מספר השורות זה 2, מספר העמודות זה 3. אם לא רשמנו אז מספר השורות והעמודות זה רגיל
- color = "r" / "b" / "y"/ .... - עבור צבעים
- plt.title("my title") - כותרת
- plt.ylabel("name of axis y") - שם ציר הווי
- plt.xlabel("name of axis x") - שם ציר האיקס
- הערה חשובה - אם זה בלי מספר גרפים בהצגה אחת אז זה יהיה:
  - plt
- אם יש מספר גרפים בהצגה אחת אז זה יהיה
  - ax[i][j]

bar plot: גרף עמודות שמשמש אותנו לבדוק כמה ערכים נמצאים בכל קטגוריה - לדוגמא אם העמודה שלנו היא מיקום אז כמה ערכים נמצאים בכל מיקום

bar plot - עושים על משתנים קאטאגוריאלים

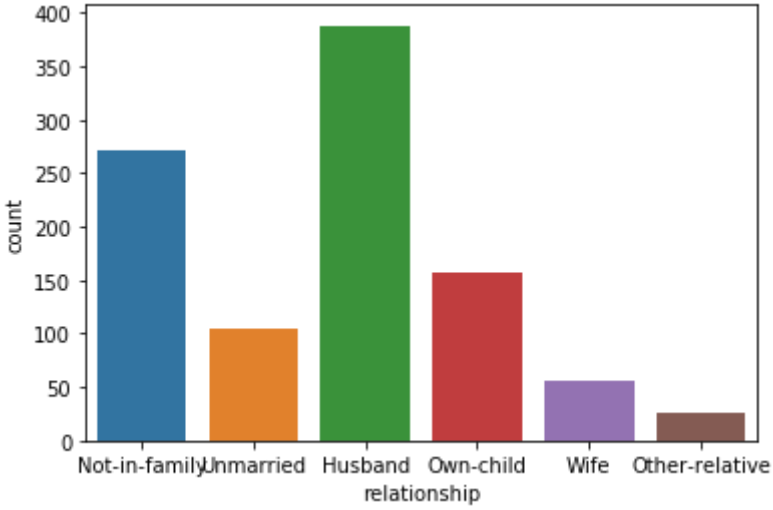
In [25]:
```
sns.countplot(x='age', data=data_sampled)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4fb40f860>
```



In [198…
```
# כמות הערכים שיש בכל relationship
sns.countplot(x='relationship', data=data_sampled)
```

Out[198…
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4842c70f0>
```

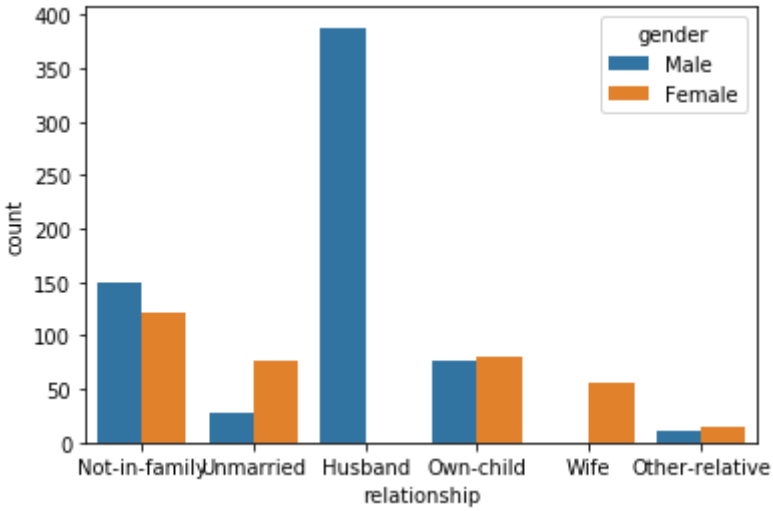


In [200…

```
# כמה ערכים יש בכל סוג קשר בחלוקה לפי מגדר
sns.countplot(x='relationship', hue='gender', data=data_sampled)
```

Out[200…

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4843897b8>
```



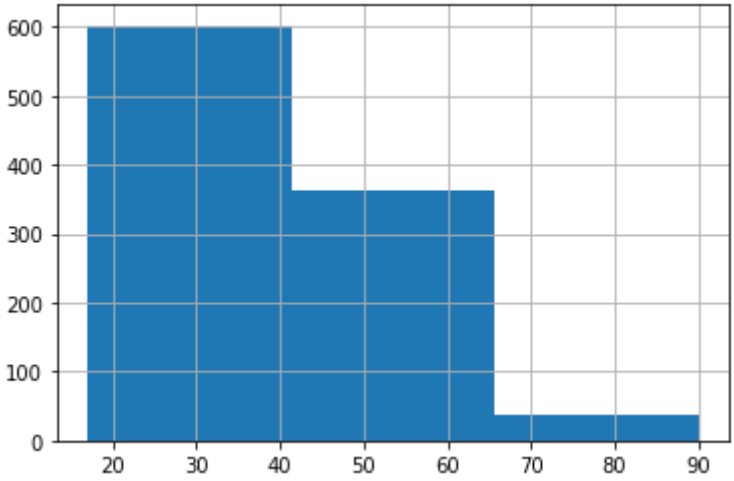
היסטוגרמה - גרף אשר בציר האיקס יהיה לנו את המשתנים שנבדוק ובציר הוואי יהיה לנו את כמות הפעמים שכל משתנה מופיע. מה היסטוגרמה שונה מגרף עמודות? בהיסטוגרמה עושים על משתנה רציף מספרי וגרף עמודות על משתנה בדיד או משתנה קטגוריאלי

In [204…

```
data_sampled.age.hist()
```

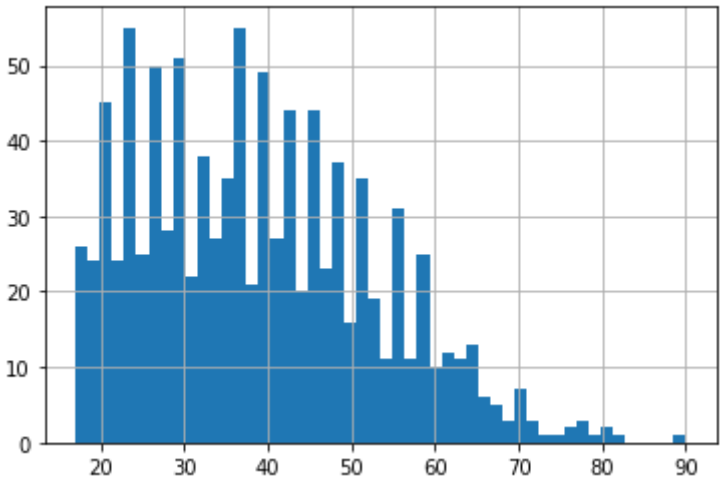
Out[204…

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4845b26d8>
```



In [296…

```
# המחשב מחשב לבד את כמות החלקים שאותם הוא מחלק
# data_sampled['age'].hist()
# bins - כמות הקבוצות
# bins - כמות הקבוצות שניצור
data_sampled['age'].hist(bins=50)
plt.show()
```



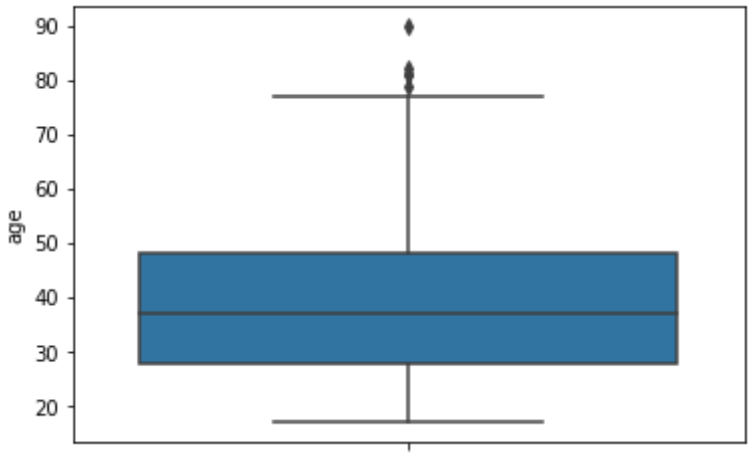
תרשים שנעשה על משתנים שהם מספרייסואז נוכל לקבל את ההתלפגות של הנתונים על - boxplot
גבי תרשים קופסא

In [34]:

```
sns.boxplot(y="age", data=data_sampled)
```

Out[34]:

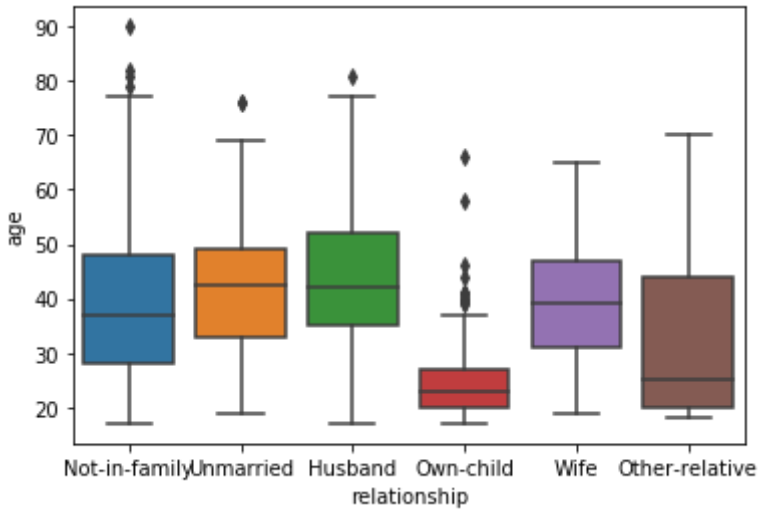
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4fc8457f0>
```



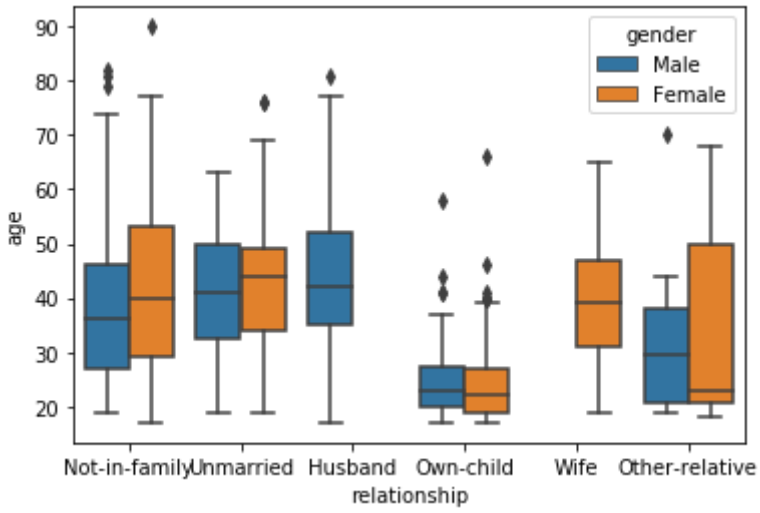
seaborn:

המשתנה של הנתונים - data לפי איזה עמודה ליצור חלוקה - hue שם העמודה - y שם העמודה - x

In [42]:
```
sns.boxplot(y='age', x='relationship', data=data_sampled)
plt.show()
```



In [43]:
```
sns.boxplot(y='age', x='relationship', hue='gender', data=data_sampled)
plt.show()
```



scatterplot

In [205…
```
url = 'https://raw.githubusercontent.com/nlihin/data-analytics/main/datasets/housing
house_df = pd.read_csv(url)
house_df.head()
```

Out[205…

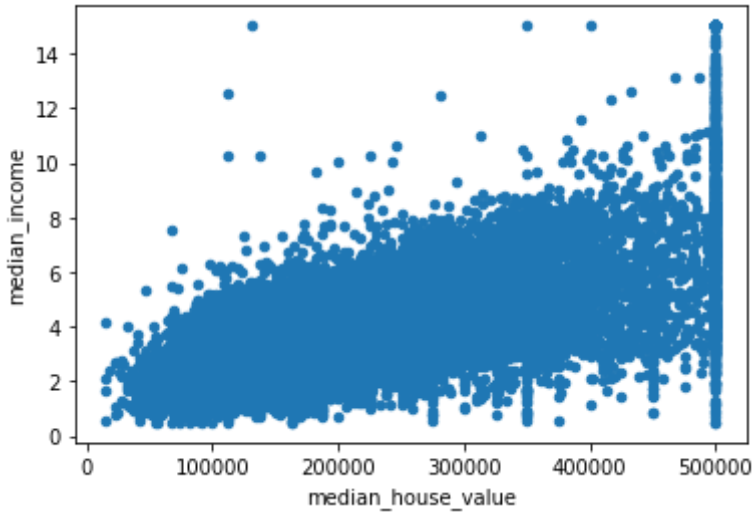
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0

In [209…

```
house_df.plot.scatter(x = 'median_house_value', y = 'median_income')
# plt.scatter(x = house_df['median_house_value'], y = house_df['median_income'])
```

Out[209…

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e48602cdd8>
```

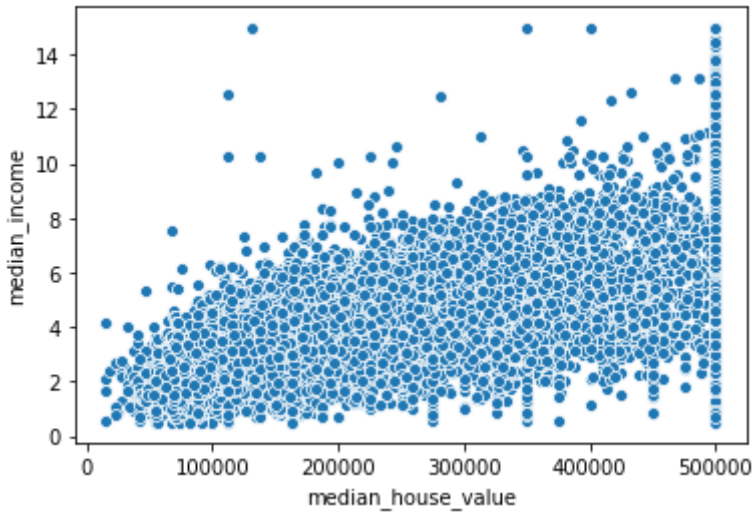


In [211…

```
sns.scatterplot(x = 'median_house_value', y = 'median_income', data=house_df)
```

Out[211…

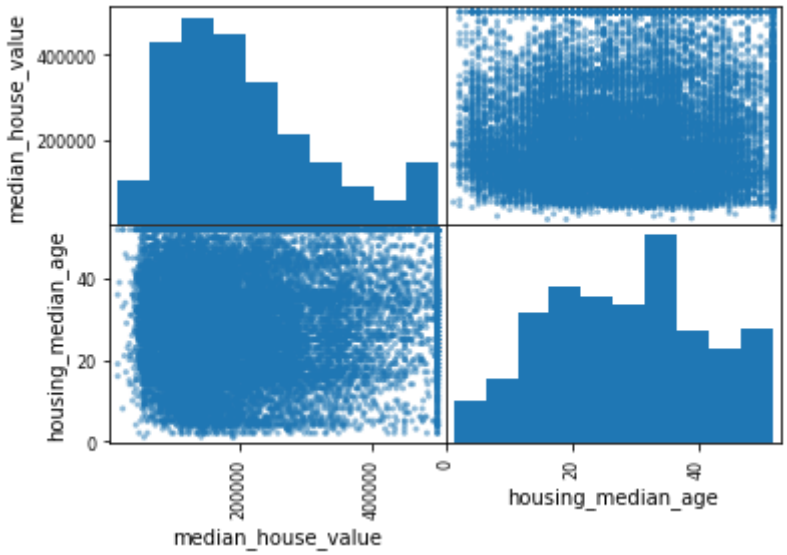
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4861899e8>
```



scatter_matrix

In [230…

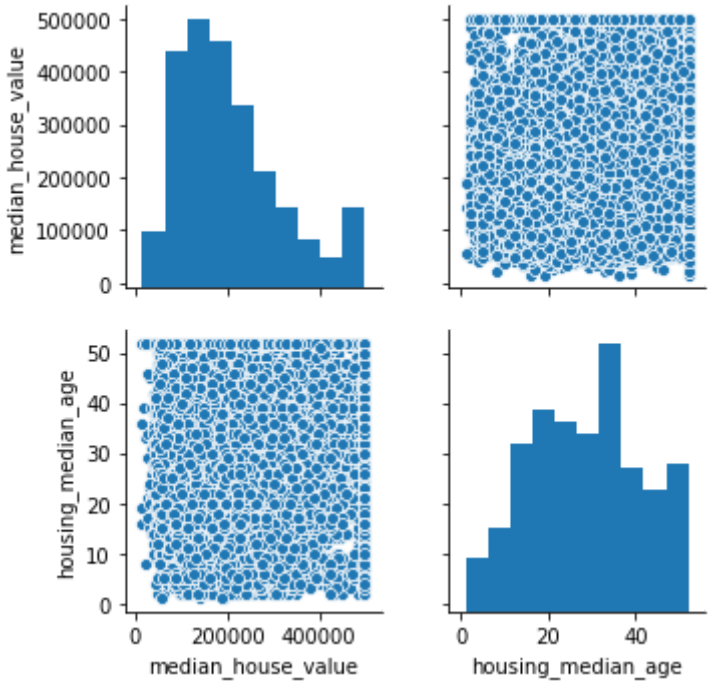
```
features = ['median_house_value', 'housing_median_age']
_ = pd.plotting.scatter_matrix(house_df[features])
```



pairplot

In [231…
```
sns.pairplot(house_df[features], height = 2.5)
```

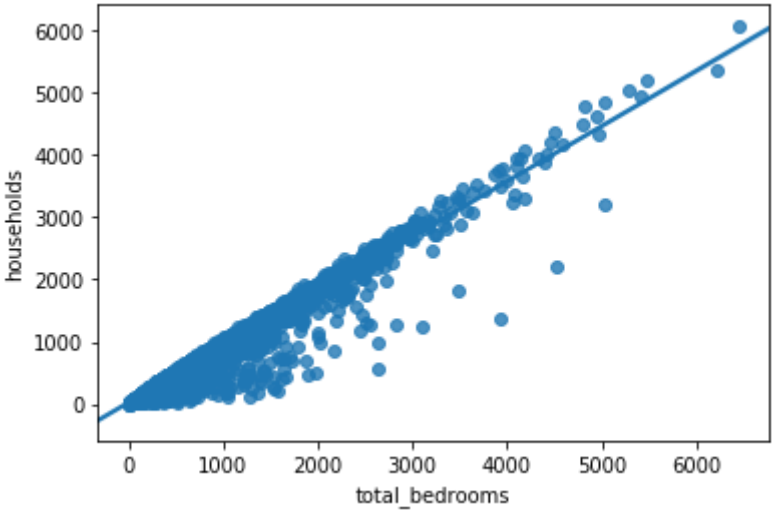
Out[231…
```
<seaborn.axisgrid.PairGrid at 0x1e489551898>
```



לבנות גרף עם קו לינארי

In [218…
```
sns.regplot(data=house_df, x='total_bedrooms', y='households')
```

Out[218…
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e487a75cc0>
```



סוגי קורלציות - קורלציה זה בדיקה של הקשר בין משתנים שונים לדוגמא האם יש קשר בין עמודה גיל לבין עמודה שכר?

- פירסון
- ספירמן

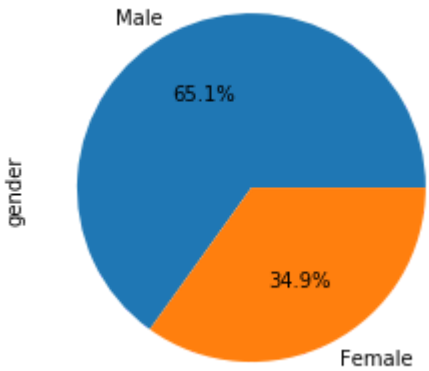
גרף עוגה

In [240…

```
data_sampled.gender.value_counts().plot.pie(autopct="%1.1f%%")
```

Out[240…

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4879577b8>
```



קורלציית פירסון נעשה בין משתנים רציפים

In [225…

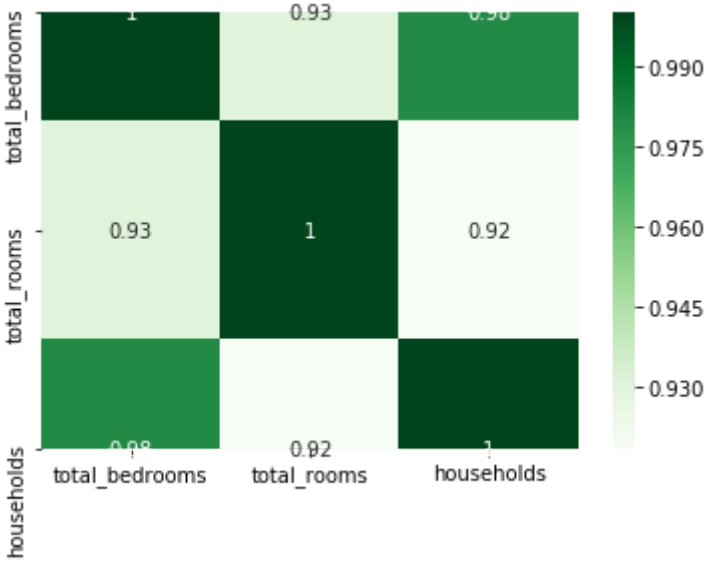
```
# קורלציה בין כל העמודות
house_df.corr(method='pearson')
# קורלציה בין חלק מהעמודות
cols = ['total_bedrooms', 'total_rooms', 'households']
corr_matrix = house_df.loc[:, cols].corr(method='pearson')
corr_matrix
```

Out[225…

	total_bedrooms	total_rooms	households
total_bedrooms	1.000000	0.930380	0.979728
total_rooms	0.930380	1.000000	0.918484
households	0.979728	0.918484	1.000000

In [226…
```
sns.heatmap(data=corr_matrix,cmap='Greens', annot=True)
```

Out[226…
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4fb80f080>
```



מודלים

In [76]:
```
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [62]:
```
iris_data = pd.read_csv("iris.csv")
iris_data.head()
print(iris_data.shape)
```

```
(150, 5)
```

In [60]:
```
X = iris_data.drop(columns='class')
X.head()
```

Out[60]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [61]:
```
y = iris_data['class']
```

In [66]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

In [67]:
```
print(X_train.shape)
print(X_test.shape)
```

```
(105, 4)
(45, 4)
```

In [75]:

```
# בחירת המודל - רק זה ישתנה, איזה סוג מודל אנחנו רוצים לבחור
model = DecisionTreeClassifier()
# תאמן את המודל על סמך הנתונים שאביא לך
model.fit(X_train, y_train)
# אחרי שאימנת תבצע חיזויים על נתונים שלא ראית עדיין
y_pred = model.predict(X_test)
# בדיקה של כמה המודל טוב
accuracy_score(y_pred, y_test)
```

Out[75]:

```
0.9111111111111111
```

In [77]:

```
# בחירת המודל - רק זה ישתנה, איזה סוג מודל אנחנו רוצים לבחור
model = RandomForestClassifier()
# תאמן את המודל על סמך הנתונים שאביא לך
model.fit(X_train, y_train)
# אחרי שאימנת תבצע חיזויים על נתונים שלא ראית עדיין
y_pred = model.predict(X_test)
# בדיקה של כמה המודל טוב
accuracy_score(y_pred, y_test)
```

```
C:\Users\Latitude e7470\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in version 0.20
to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Out[77]:

```
0.9333333333333333
```

דוגמא נוספת על טיטאניק

In [250…

```
# data_titanic = pd.read_csv('titanic_train.csv')
# data_titanic.head()
```

In [251…

```
# data_titanic = data_titanic.replace({'male':0, 'female':1})
# data_titanic.head()
```

פעולות שנרצה לעשות:

- חלוקה של הנתונים
- יצירת המודל
- אימון המודל
- חיזוי
- בדיקת טיב המודל

In [252…

```
iris_data.head()
```

Out[252…

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa

	sepal_length	sepal_width	petal_length	petal_width	class
4	5.0	3.6	1.4	0.2	Iris-setosa

In [254…

```
iris_data.isnull().sum()
```

Out[254…

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
class           0
dtype: int64
```

In [258…

```
# יצירה של X ו Y
features = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
X = iris_data.loc[:, features]
y = iris_data['class']
# חלוקה של הנתונים ל:
# train and test
X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split(X, y, te
# עכשיו נרצה ליצור את המודל
model = sklearn.tree.DecisionTreeClassifier()
# לאמן את המודל
model.fit(X_train, y_train)
# חיזוי
y_pred = model.predict(X_test)
# בדיקה
accuracy_score(y_pred, y_test)
```

Out[258…

```
0.9555555555555556
```

In [262…

```
avg = data_titanic.Age.mean()
data_titanic['Age'] = data_titanic.Age.fillna(avg)
data_titanic.isnull().sum()
data_titanic.head()
```

Out[262…

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	En
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	C123	

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	En
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	NaN	

ננסה להריץ על הטיטאניק

In [264…

```
# יצירה של X ו Y
features = ['Age', 'Parch', 'Pclass', 'Sex']
X = data_titanic.loc[:, features]
y = data_titanic['Survived']
# חלוקה של הנתונים ל:
# train and test
X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split(X, y, te
# עכשיו נרצה ליצור את המודל
model = sklearn.tree.DecisionTreeClassifier()
# לאמן את המודל
model.fit(X_train, y_train)
# חיזוי
y_pred = model.predict(X_test)
# בדיקה
accuracy_score(y_pred, y_test)
```

Out[264…

```
0.7611940298507462
```

איך נדע האם המודל יותר טוב מרנדום?

In [265…

```
data_titanic.Survived.value_counts().plot.pie(autopct="%1.1f%%")
```

Out[265…

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4f2157a58>
```