سوال ۱: برای هریک از موارد زیر یک cube ایجاد نمایید:

ابتدا کتابخانه Pandas را import کرده و سپس فایل اکسل را در پایتون بارگذاری می کنیم و برای هریک از موارد زیر cube را می سازیم

- **میانگین درآمد افراد هر شهر در ماه**

برای این کوئری ابتدا ستون های CountyName و Daramad_Total_Rials را جدا کرده و سپس روی ستونCountyName هر شهر را با استفاده از groupby جدا می کنیم و از ()mean برای گرفتن میانگین در آمد در هر شهر استفاده می کنیم و نام ستون به میانگین درآمد تغییر می کند که بصورت زیر است :

In [2]:
```
import numpy as np
import pandas as pd
df=pd.read_csv("F:\\IUST\\داده کاوی\\ترم 2\\DataMining_HW2\\500000FamilySample-990402\\500000FamilySample-990402.csv")
```

Question 1.1

In [12]:
```
cf=df[['CountyName','Daramad_Total_Rials']]
cf1=cf.groupby('CountyName').mean()
cf1.rename(columns={'Daramad_Total_Rials':'Daramad_mean_Rials'},inplace=True)
cf1
```

Out[12]:

CountyName	Daramad_mean_Rials
آبادان	5.916593e+06
آباده	6.939481e+06
آبدانان	1.670656e+06
آبیک	6.661873e+06
آذرشهر	4.894767e+06
...	...
گناوه	4.101024e+06
گنبدکاووس	3.106371e+06
گچساران	6.647401e+06
گیلانغرب	5.911495e+06
یزد	8.453853e+06

422 rows × 1 columns

- **تعداد اعضای دارای بیماری خاص استان تهران به تفکیک شهر و وضعیت بهزیستی**

برای این کوئری ابتدا ستون های 'ProvinceName' , 'CountyName' , 'IsBehzisti_Malool' , 'IsBimarkhas' را جدا کرده و سپس روی ستون استان , استان تهران و روی ستون بیماری خاص عدد ۱ را فیلتر میکنیم و سطرهای با استان تهران و دارای بیماری خاص را انتخاب می کنیم. سپس روی ۲ ستون 'CountyName' , 'IsBehzisti_Malool' عمل groupby را انجام داده و به تفکیک شهر و وضعیت بهزیستی گروه بندی می کنیم و عمل sum() را در هر گروه برای مقادیر بیماری خاص اعمال می کنیم تا تعداد اعضای دارای بیماری خاص مشخص شوند.

کد و خروجی:

Question 1.2

In [16]:
```
pf=df[['IsBimarkhas','IsBehzisti_Malool','CountyName','ProvinceName']]
pf1=pf.loc[pf.ProvinceName == 'تهران']
pf1.loc[pf1.IsBimarkhas == 1]
pf2 = pf1.groupby(['IsBehzisti_Malool','CountyName']).sum()
pf3=pf2[['IsBimarkhas']]
pf3.rename(columns={'IsBimarkhas':'Count_Bimarkhas'},inplace=True)
pf3.unstack()
```

Out[16]:

	Count_Bimarkhas																
CountyName	اسلامشهر	بهارستان	تهران	دماوند	رباطکریم	ری	شمیرانات	شهریار	فیروزکوه	قدس	قرچک	ملارد	ورامین	پاکدشت	پردیس	پیشوا	
IsBehzisti_Malool																	
0	17	6	479	4	10	11	6	22	1	6	1	8	14	14	3	1	
1	1	3	58	1	3	8	5	0	0	1	0	3	2	1	0	0	

- **تعداد پروانه های صنفی افراد شهرنشین در استان مازندران به تفکیک شهر و سال تولد**

برای این کوئری ابتدا ستون های 'ProvinceName','CountyName','SenfName','BirthDate','IsUrban' را جدا کرده و سپس رکورد هایی که در استان مازندران است و وضعیت شهرنشینی عدد ۱ است را فیلتر می کنیم و سپس سال تولد را از ستون تاریخ تولد جدا می کنیم . سپس روی ۲ ستون 'CountyName' , 'Year' عمل groupby را انجام داده و به تفکیک نام شهر و سال تولد گروه بندی می کنیم و عمل count() را در هر گروه برای Senfname اعمال می کنیم تا تعداد پروانه صنفی هر گروه مشخص شود.

Question 1.3

In [24]:
```
gf=df[['ProvinceName','CountyName','SenfName','BirthDate','IsUrban']]
nf = df.loc[df.ProvinceName == 'مازندران']
nf2 = nf.loc[nf.IsUrban == 1]
nf2['Year']=pd.DatetimeIndex(nf2['BirthDate']).year
nf3 = nf2.groupby(['CountyName' , 'Year']).count()
nf4=nf3[['SenfName']]
nf4.rename(columns={'SenfName':'Count_Senf'},inplace=True)
nf4
#nf4.unstack()
```

Out[24]:

		Count_Senf
CountyName	**Year**	
آمل	**1922**	0
	1923	0
	1924	0
	1925	0
	1926	0
...	...	...
گلوگاه	**2014**	0
	2015	0
	2016	0
	2017	0
	2018	0

1702 rows × 1 columns

- **مقدار واریزی افراد به تفکیک سال و استان و جنسیت و وضعیت شهرنشینی**

برای این کوئری ابتدا ستون های 'Variz95','Variz96','Variz97', 'ProvinceName' , 'Gender', 'IsUrban' را جدا کرده و سپس روی ستون های 'ProvinceName' , 'IsUrban', 'Gender' عمل groupby را انجام داده و به تفکیک استان و جنسیت و شهرنشینی گروه بندی می کنیم و عمل sum()را برای هر گروه انجام می دهیم تا مجموع واریزی های هر سال برای هر گروه مشخص شود . کد و خروجی:

In [17]:
```
kf1=df[['Variz95' , 'ProvinceName' , 'Gender', 'IsUrban']]
kf2 = kf1.groupby(['ProvinceName' , 'IsUrban', 'Gender']).sum()
kf2.unstack()
```

Out[17]:

		Variz95	
	Gender	زن	مرد
ProvinceName	**IsUrban**		
آذربایجان شرقی	**0**	690997500000	5607597500000
	1	7256835000000	33655485000000
آذربایجان غربی	**0**	705227500000	6229060000000
	1	4319217500000	21165382500000
اردبیل	**0**	203032500000	1664590000000
...	...	...	...
گلستان	**1**	2583012500000	7070132500000
گیلان	**0**	931777500000	3860947500000
	1	5327442500000	15891052500000
یزد	**0**	41260000000	189052500000
	1	494467500000	1375540000000

62 rows × 2 columns

- **میانگین قیمت ماشین های هر خانواده به تفکیک شهر و تعداد اعضای خانواده**

برای این کوئری ابتدا ستون های 'Id' ,'ParentId' , 'CountyName' , 'Cars_Count', 'CarPrice_Sum' را جدا می کنیم.
سپس میانگین قیمت ماشین های هر شخص را حساب می کنیم(qf).
سپس روی qf عمل groupby را روی ستون های 'CountyName' ,'ParentId' انجام داده و عمل Count() را انجام می دهیم تا تعداد اعضای هر خانواده در ستون Id ثبت می شود.(qf2)

یک بار دیگر روی qf عمل groupby را روی ستون های 'CountyName' ,'ParentId' انجام داده و عمل Sum() را انجام می دهیم تا مجموع میانگین قیمت ماشین های هر شخص در خانواده حساب شود که در ستون mean_car ثبت می شود.(qf4)

سپس ۲ جدول qf2 و qf4 را با هم join می کنیم. و در آخر میانگین قیمت ماشین های هر خانواده را با تقسیم کردن ستون mean_car بر تعداد اعضای خانواده بدست می آوریم و تعداد اعضای خانواده هم با توجه به ۲ خط بالاتر که در Id بود به نام count_family number تغییر می دهیم.(qf5)

با توجه به آنکه در صورت سوال ذکر شده به تفکیک تعداد اعضای خانواده هم باشد در آخر با اعمال groupby() روی ستون شهر و تعداد اعضای خانواده و انجام عمل mean() روی ستون mean_all میانگین قیمت ماشین های هر خانواده به تفکیک شهر و تعداد اعضای خانواده بدست می آید(qf6) کد و خروجی:

Question 1.5

In [32]:
```
qf = pd.DataFrame(df , columns=['Id' ,'ParentId' , 'CountyName' , 'Cars_Count', 'CarPrice_Sum'])
qf['mean_car'] = qf['CarPrice_Sum'] / qf['Cars_Count']

qf1 = qf.groupby([ 'CountyName' ,'ParentId']).count()
qf2 = qf1[['Id']]

qf3 = qf.groupby([ 'CountyName' ,'ParentId']).sum()
qf4 = qf3[['mean_car']]

qf5 = pd.merge(qf2, qf4, on=["CountyName", "ParentId"])
qf5['mean_all'] = qf5['mean_car'] / qf5['Id']
qf5.pop('mean_car')
qf5.rename(columns={'Id':'Count_family members'},inplace=True)

qf6=qf5.groupby([ 'CountyName','Count_family members']).mean()
qf6

#qf5.unstack()
```

Out[32]:

		mean_all
CountyName	Count_family members	
آبادان	1	1.471978e+08
	2	9.750172e+07
	3	1.032106e+08
	4	9.866679e+07
	5	7.983862e+07
...	...	...
یزد	4	1.785588e+08
	5	1.248261e+08
	6	9.678180e+07
	7	1.027846e+08
	9	5.586361e+07

2871 rows × 1 columns

سوال۲: ۵ مورد cube دیگر با ذکر دلیل :

- **میزان خرید در ماه های مختلف سال ۹۸ به تفکیک جنسیت و ماه**

دلیل: با توجه بی این کوئری می توان ماه هایی که خرید کمتری صورت می گیرد را دریافت و با توجه به جنسیت کالاهای مربوطه را با تخفیف در آن ماه ها عرضه کرد تا قدرت خرید بالاتر رود.

توضیحات: ابتدا ستون های 'Gender','Card9801','Card9802','Card9803','Card9804','Card9805','Card9806' را جدا می کنیم سپس با اعمال groupby() به تفکیک جنسیت گروه بندی می کنیم و برای هر گروه عمل sum() را انجام می دهیم تابرای ستون هر ماه میزان خرید بدست آید.

کد و خروجی:

Question 2.1

```
In [33]: wf=df[['Gender','Card9801','Card9802','Card9803','Card9804','Card9805','Card9806']]
         wf1=wf.groupby('Gender').sum()
         wf1
```

Out[33]:

Gender	Card9801	Card9802	Card9803	Card9804	Card9805	Card9806
زن	7028723980639	10583967959319	9310069867321	10382216669487	10277872042906	9911191591783
مرد	26414716768290	40451734293512	34056790848498	37647962919552	36978359860629	35397861856161

- **میانگین درآمد بازنشستگان دارای بیماری خاص به تفکیک جنسیت و استان**

دلیل: با توجه به این کوئری می توان درآمد بازنشستگانی که بیماری خاص دارند را در استان های مختلف برآورد کرد و در صورت نیاز مبلغی را جهت درمان به حقوق آن ها اضافه نمود.

توضیحات: ابتدا ستون های
'Id','Gender','ProvinceName','Daramad_Total_Rials','IsBazneshaste_Sandoghha','IsBimarkhas' را جدا کرده سپس رکورد هایی که بیماری خاص و بازنشستگی برای آن ها عدد ۱ هست را فیلتر می کنیم.

عمل groupby روی ۲ ستون 'ProvinceName','Gender' انجام می دهیم و برای هر گروه میانگین درآمد را حساب می کنیم.

کد و خروجی:

Question 2.2

```
In [41]: uf=df[['Id','Gender','ProvinceName','Daramad_Total_Rials','IsBazneshaste_Sandoghha','IsBimarkhas']]
         uf1 = uf.loc[uf.IsBazneshaste_Sandoghha == 1]
         uf2 = uf1.loc[uf1.IsBimarkhas == 1]
         uf3=uf2.groupby(['ProvinceName','Gender']).mean()
         uf3.rename(columns={'Daramad_Total_Rials':'Mean_Daramad'},inplace=True)
         uf4=uf3[['Mean_Daramad']]
         uf4.unstack()
```

Out[41]:

	Mean_Daramad	
Gender	زن	مرد
ProvinceName		
آذربایجان شرقی	1.556876e+07	2.592299e+07
آذربایجان غربی	1.531047e+07	2.000210e+07
اردبیل	7.893211e+06	2.134237e+07
اصفهان	1.997251e+07	2.203604e+07
البرز	1.206732e+07	1.497351e+07
بوشهر	NaN	1.033834e+07
تهران	1.855023e+07	2.001569e+07
خراسان جنوبی	NaN	1.993546e+07
خراسان رضوی	1.885855e+07	1.992535e+07
خراسان شمالی	1.382816e+07	1.179796e+07
خوزستان	1.751985e+07	2.271651e+07
زنجان	2.855102e+07	1.547660e+07
سمنان	1.516881e+07	2.937604e+07
فارس	1.850054e+07	1.978961e+07
قزوین	2.154518e+07	2.912634e+07
قم	1.257841e+07	1.191353e+07
مازندران	1.637690e+07	1.978185e+07
مرکزی	1.604639e+07	2.431791e+07
همدان	NaN	2.166973e+06
چهارمحال وبختیاری	3.014542e+07	1.516881e+07
کرمان	2.058288e+07	2.471735e+07
کهگیلویه و بویراحمد	NaN	4.906283e+07
گلستان	1.807980e+07	1.974839e+07
گیلان	1.881491e+07	2.273996e+07
یزد	NaN	2.000000e+07

- **تعداد سفرغیر زیارتی هوایی به تفکیک سال و جنسیت و استان**

دلیل: با توجه به این کوئری می توان به روند ســـفر های غیر زیارتی هوایی پی برد و با توجه به آن از شـــرکت های هواپیمایی در شهرهای مختلف مالیات دریافت کرد و ضمن آن شرکت های هواپیمایی برای تبلیغ یا عدم تبلیغ این سفرها و کنترل قیمت این سفرها در شهرهای مختلف برنامه ریزی کنند.

توضیحات: ابتدا ستون های
Id','ProvinceName','Gender','Trip_AirNonPilgrimageCount_95','Trip_AirNonPilgrimageCount_96''
,'Trip_AirNonPilgrimageCount_97','Trip_AirNonPilgrimageCount_98

را جدا می کنیم سپس عمل groupby را روی ستون های 'ProvinceName','Gender' انجام می دهیم و سپس برای هر گروه عمل sum() را انجام می دهیم تا مجموع سفر های هر گروه برای هر سال در ستون های مختلف بدست آید.

کد و خروجی:

Question 2.3

```
In [22]: ef=df[['Id','ProvinceName','Gender','Trip_AirNonPilgrimageCount_95','Trip_AirNonPilgrimageCount_96','Trip_AirNonPilgrimageCount_9

ef1=ef.groupby(['ProvinceName','Gender']).sum()
ef1.pop('Id')
ef1
```

Out[22]:

ProvinceName	Gender	Trip_AirNonPilgrimageCount_95	Trip_AirNonPilgrimageCount_96	Trip_AirNonPilgrimageCount_97	Trip_AirNonPilgrimageCount_98
آذربایجان شرقی	زن	356	601	642	394
	مرد	472	817	857	481
آذربایجان غربی	زن	169	221	261	175
	مرد	226	318	351	235
اردبیل	زن	31	40	66	35
...	...	...	...	...	...
گلستان	مرد	138	222	226	149
گیلان	زن	227	430	386	357
	مرد	232	377	344	278
یزد	زن	12	19	10	3
	مرد	6	12	13	4

62 rows × 4 columns

- **میزان برداشت به تفکیک سال و استان و وضعیت شهرنشینی**

دلیل: با توجه به این کوئری می توان میزان خروج پول را در سال های مختلف و در شهر و روستاها برآورد کرد و برای جلوگیری از خروج بیش از حد سرمایه ها از بانک ها برنامه ریزی نمود(مثلا افزایش سود)

توضیحات: ابتدا ستون های 'Id','ProvinceName','IsUrban','Bardasht95','Bardasht96','Bardasht97' را جدا می کنیم سپس با عمل groupby را روی ستون های 'ProvinceName','IsUrban' انجام می دهیم و برای هر گروه عمل sum() را انجام می دهیم تا مجموع برداشت ها به تفکیک استان و وضعیت شهر نشینی برای هر گروه محاسبه شود.

Question 2.4

کد و خروجی:

```
In [30]: yf=df[['Id','ProvinceName','IsUrban','Bardasht95','Bardasht96','Bardasht97']]
yf1=yf.groupby(['ProvinceName','IsUrban']).sum()
yf1.pop('Id')
yf1
```

Out[30]:

ProvinceName	IsUrban	Bardasht95	Bardasht96	Bardasht97
آذربایجان شرقی	0	6244960000000	7862732500000	17393762500010
	1	40910500000000	48594430000000	120359317500113
آذربایجان غربی	0	6836050000000	8584257500000	21168617500015
	1	25405337500000	30522922500000	78249902500066
اردبیل	0	1846057500000	2474970000000	5169985000004
...	...	...	...	...
گلستان	1	9661002500000	12163295000000	28049240000024
گیلان	0	4743727500000	5911082500000	13373142500012
	1	21149475000000	25792515000000	61342370000045
یزد	0	225755000000	285125000000	586922500001
	1	1855225000000	2304730000000	5407860000005

62 rows × 3 columns

- **تعداد کارگرهای کارفرماها به تفکیک استان و وضعیت شهرنشینی**

دلیل: با توجه به این کوئری می توان میزان فعالیت های عمرانی در شـهر و روسـتاهای اسـتان های مختلف را برآورد کرد و همچنین تعداد کارگر ها و اقشار ضعیف تر در استان های مختلف را نیز برآورد کرد.

توضـیحات: ابتدا سـتون های 'ProvinceName','IsUrban','IsTamin_Karfarma','Tamin_KargarCount' را جدا می کنیم سـپس رکوردهای کارفرماها را که سـتون 'IsTamin_Karfarma' برای آنها عدد ۱ اسـت را فیلتر می کنیم و سـپس با سـتون های 'ProvinceName','IsUrban' عمل groupby را انجام می دهیم و سـپس برای هر گروه با اعمال sum() تعداد کارگر ها را محاسبه می کنیم

کد و خروجی:

Question 2.5

```
In [57]: zf=df[['ProvinceName','IsUrban','IsTamin_Karfarma','Tamin_KargarCount']]
         zf1 = zf.loc[zf.IsTamin_Karfarma == 1]
         zf1=zf.groupby(['ProvinceName','IsUrban']).sum()
         zf1.pop('IsTamin_Karfarma')
         zf1.unstack()
```

Out[57]:

	Tamin_KargarCount	
IsUrban	0	1
ProvinceName		
آذربایجان شرقی	450	2977
آذربایجان غربی	338	1326
اردبیل	294	1025
اصفهان	609	5537
البرز	29	603
ایلام	0	0
بوشهر	8	23
تهران	666	14159
خراسان جنوبی	93	240
خراسان رضوی	905	3537
خراسان شمالی	72	386
خوزستان	851	2609
زنجان	2	79
سمنان	6	52
سیستان و بلوچستان	11	74
فارس	428	3257
قزوین	12	102
قم	35	629
لرستان	3	47
مازندران	1514	2564
مرکزی	126	1845
هرمزگان	15	271
همدان	1	181
چهارمحال وبختیاری	41	31
کردستان	5	30
کرمان	370	1534
کرمانشاه	14	64
کهگیلویه و بویراحمد	11	30
گلستان	281	658
گیلان	316	1713
یزد	16	109