In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [10]:
```
df=pd.read_csv('honeyproduction.csv')
```

In [11]:
```
df.head()
```

Out[11]:

	state	numcol	yieldpercol	totalprod	stocks	priceperlb	prodvalue	year
0	AL	16000.0	71	1136000.0	159000.0	0.72	818000.0	1998
1	AZ	55000.0	60	3300000.0	1485000.0	0.64	2112000.0	1998
2	AR	53000.0	65	3445000.0	1688000.0	0.59	2033000.0	1998
3	CA	450000.0	83	37350000.0	12326000.0	0.62	23157000.0	1998
4	CO	27000.0	72	1944000.0	1594000.0	0.70	1361000.0	1998

In [17]:
```
df.describe(include='all').transpose()
```

Out[17]:

	count	unique	top	freq	mean	std	min	25%	
state	626	44	AL	15	NaN	NaN	NaN	NaN	
numcol	626.0	NaN	NaN	NaN	60284.345048	91077.087231	2000.0	9000.0	260
yieldpercol	626.0	NaN	NaN	NaN	62.009585	19.458754	19.0	48.0	
totalprod	626.0	NaN	NaN	NaN	4169086.261981	6883846.751268	84000.0	475000.0	15330
stocks	626.0	NaN	NaN	NaN	1318859.42492	2272963.665923	8000.0	143000.0	4395
priceperlb	626.0	NaN	NaN	NaN	1.409569	0.638599	0.49	0.9325	
prodvalue	626.0	NaN	NaN	NaN	4715741.214058	7976109.76856	162000.0	759250.0	18415
year	626.0	NaN	NaN	NaN	2004.864217	4.317306	1998.0	2001.0	20

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 626 entries, 0 to 625
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   state        626 non-null    object 
 1   numcol       626 non-null    float64
 2   yieldpercol  626 non-null    int64  
 3   totalprod    626 non-null    float64
 4   stocks       626 non-null    float64
 5   priceperlb   626 non-null    float64
 6   prodvalue    626 non-null    float64
 7   year         626 non-null    int64  
dtypes: float64(5), int64(2), object(1)
memory usage: 39.2+ KB
```

In [18]:
```
df['state'].nunique()
```

Out[18]:
```
44
```

In [22]:
```
df['state'].value_counts()
```

Out[22]:
```
AL    15
SD    15
NJ    15
NM    15
NY    15
NC    15
ND    15
OH    15
OR    15
PA    15
TN    15
MT    15
TX    15
UT    15
VT    15
VA    15
WA    15
WV    15
WI    15
WY    15
AZ    15
NE    15
MO    15
IN    15
AR    15
CA    15
CO    15
FL    15
GA    15
HI    15
ID    15
IL    15
IA    15
KS    15
KY    15
LA    15
ME    15
MI    15
MN    15
MS    15
NV    11
OK     6
MD     6
SC     3
Name: state, dtype: int64
```

In [23]:
```
df['state'].unique()
```

Out[23]:
```
array(['AL', 'AZ', 'AR', 'CA', 'CO', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN',
       'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MI', 'MN', 'MS', 'MO', 'MT',
       'NE', 'NV', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA',
       'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY', 'SC'],
      dtype=object)
```

In [24]:
```
df[['state','totalprod']].groupby('state').mean().round()
```

Out[24]:

	totalprod
state	
AL	825467.0
AR	2810400.0
AZ	2032267.0
CA	23169000.0
CO	1750600.0
FL	16469867.0
GA	3299933.0
HI	843133.0
IA	2080000.0
ID	4410667.0
IL	498333.0
IN	484000.0
KS	707933.0
KY	229667.0
LA	3627333.0
MD	211000.0
ME	246733.0
MI	4854667.0
MN	9624000.0
MO	871533.0
MS	1456867.0
MT	10437467.0
NC	542733.0
ND	31672333.0
NE	3158600.0
NJ	399533.0
NM	476467.0
NV	439273.0
NY	3937467.0
OH	1040067.0
OK	201167.0
OR	2121400.0
PA	1295600.0

	totalprod
state	
SC	343333.0
SD	17742733.0
TN	407733.0
TX	6993600.0
UT	1179067.0
VA	266533.0
VT	388067.0
WA	2687733.0
WI	5455533.0
WV	321200.0
WY	2617933.0

```
In [25]: df['year'].nunique()
```

```
Out[25]: 15
```

```
In [26]: df['year'].min()
```

```
Out[26]: 1998
```

```
In [27]: df['year'].max()
```

```
Out[27]: 2012
```

```
In [28]: df[df['totalprod']==df['totalprod'].max()]
```

Out[28]:

	state	numcol	yieldpercol	totalprod	stocks	priceperlb	prodvalue	year
532	ND	510000.0	91	46410000.0	12995000.0	1.5	69615000.0	2010

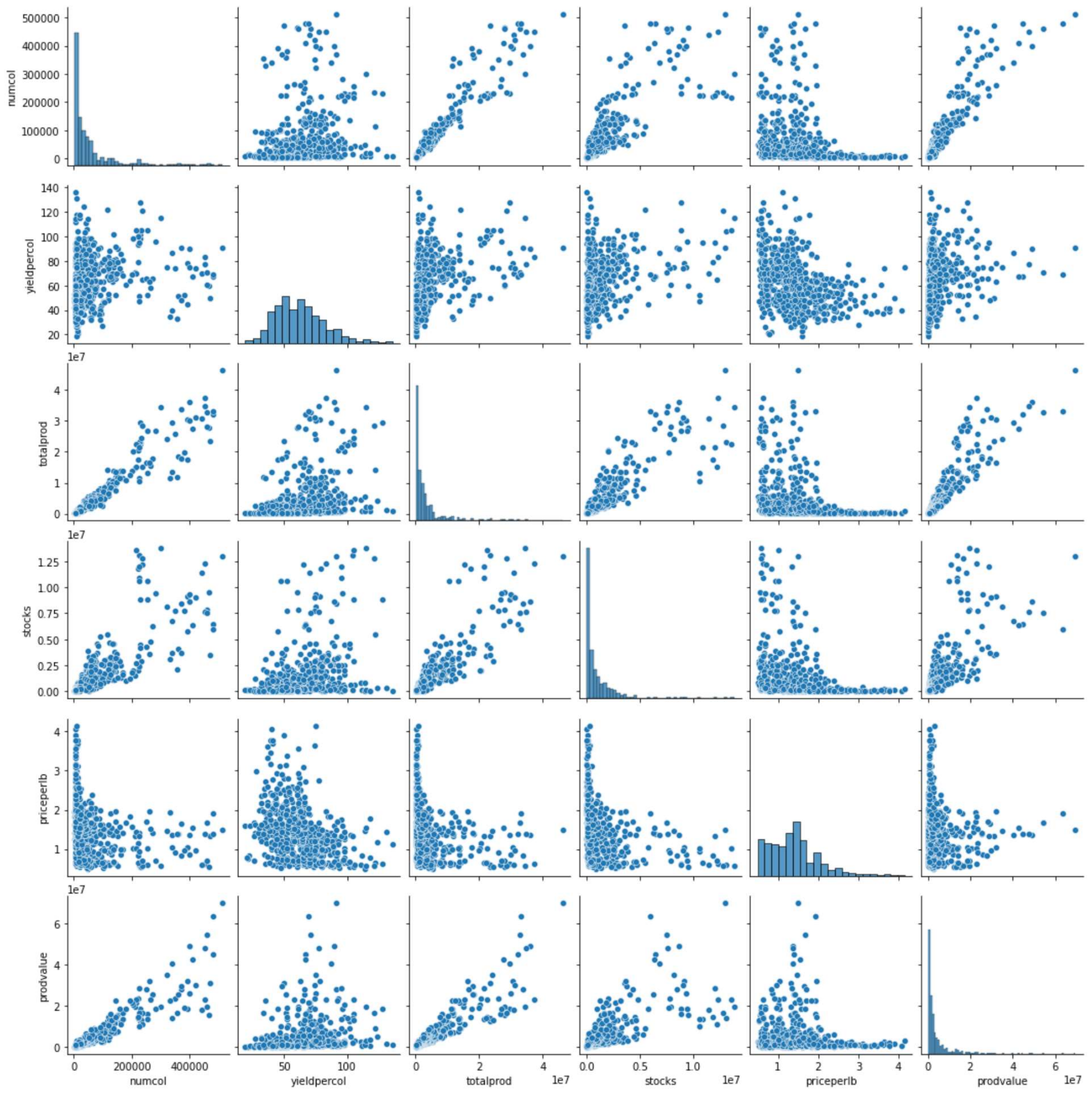
In [29]:
```
df[['year','yieldpercol']].groupby('year').mean().round()
```

Out[29]:

	yieldpercol
year	
1998	70.0
1999	65.0
2000	68.0
2001	65.0
2002	67.0
2003	63.0
2004	65.0
2005	64.0
2006	62.0
2007	59.0
2008	61.0
2009	54.0
2010	56.0
2011	55.0
2012	55.0

In [30]:
```
sns.pairplot(df[['numcol','yieldpercol','totalprod','stocks','priceperlb','prodva
```

Out[30]:
```
<seaborn.axisgrid.PairGrid at 0x1c75aadbaf0>
```



In [32]:
```
cor=df[['numcol',
       'yieldpercol',
       'totalprod',
       'stocks',
       'priceperlb',
       'prodvalue']].corr()
cor
```

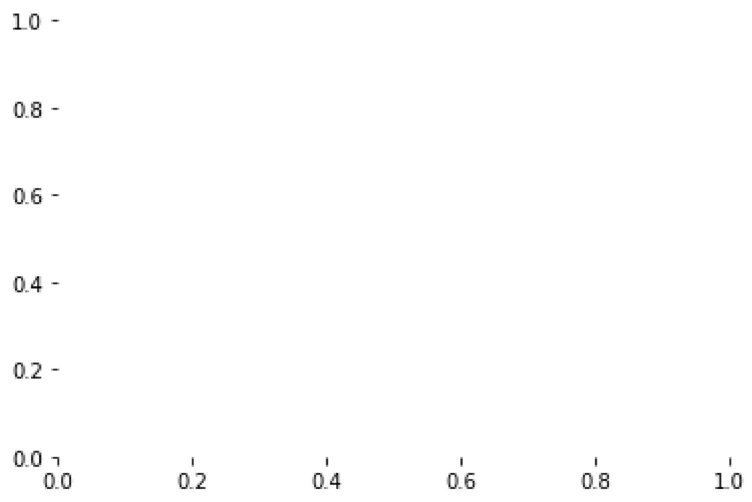
Out[32]:

	numcol	yieldpercol	totalprod	stocks	priceperlb	prodvalue
numcol	1.000000	0.243515	0.953594	0.825929	-0.232701	0.912796
yieldpercol	0.243515	1.000000	0.396252	0.367812	-0.358646	0.278977
totalprod	0.953594	0.396252	1.000000	0.878830	-0.264499	0.907236
stocks	0.825929	0.367812	0.878830	1.000000	-0.305867	0.728560
priceperlb	-0.232701	-0.358646	-0.264499	-0.305867	1.000000	-0.089567
prodvalue	0.912796	0.278977	0.907236	0.728560	-0.089567	1.000000

```
In [34]: sns.heatmap(cor,annot=True,camp='plasma',vmin=-1,vmax=1)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_13040/2712056110.py in <module>
----> 1 sns.heatmap(cor,annot=True,camp='plasma',vmin=-1,vmax=1)

~\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_decorator
s.py in inner_f(*args, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)
})
---> 46         return f(**kwargs)
     47     return inner_f
     48 

~\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\matrix.py
 in heatmap(data, vmin, vmax, cmap, center, robust, annot, fmt, annot_kws, li
newidths, linecolor, cbar, cbar_kws, cbar_ax, square, xticklabels, yticklabel
s, mask, ax, **kwargs)
    551     if square:
    552         ax.set_aspect("equal")
--> 553     plotter.plot(ax, cbar_ax, kwargs)
    554     return ax
    555 

~\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\matrix.py
 in plot(self, ax, cax, kws)
    300 
    301         # Draw the heatmap
--> 302         mesh = ax.pcolormesh(self.plot_data, cmap=self.cmap, **kws)
    303 
    304         # Set the axis limits

~\AppData\Local\Programs\Python\Python39\lib\site-packages\matplotlib\__init_
_.py in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413 
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python39\lib\site-packages\matplotlib\axes\_a
xes.py in pcolormesh(self, alpha, norm, cmap, vmin, vmax, shading, antialiase
d, *args, **kwargs)
   6022         kwargs.setdefault('snap', rcParams['pcolormesh.snap'])
   6023 
-> 6024         collection = mcoll.QuadMesh(
   6025             coords, antialiased=antialiased, shading=shading,
   6026             array=C, cmap=cmap, norm=norm, alpha=alpha, **kwargs)

~\AppData\Local\Programs\Python\Python39\lib\site-packages\matplotlib\collect
ions.py in __init__(self, *args, **kwargs)
   2012         # super init delayed after own init because array kwarg requi
res
   2013         # self._coordinates and self._shading
```

```
-> 2014         super().__init__(**kwargs)
   2015
   2016     # Only needed during signature deprecation

~\AppData\Local\Programs\Python\Python39\lib\site-packages\matplotlib\collect
ions.py in __init__(self, edgecolors, facecolors, linewidths, linestyles, cap
style, joinstyle, antialiaseds, offsets, transOffset, norm, cmap, pickradius,
hatch, urls, zorder, **kwargs)
    219
    220         self._path_effects = None
--> 221         self.update(kwargs)
    222         self._paths = None
    223

~\AppData\Local\Programs\Python\Python39\lib\site-packages\matplotlib\artist.
py in update(self, props)
   1062                 func = getattr(self, f"set_{k}", None)
   1063                 if not callable(func):
-> 1064                     raise AttributeError(f"{type(self).__name__!
r} object "
   1065                                          f"has no property {k!
r}")
   1066                 ret.append(func(v))

AttributeError: 'QuadMesh' object has no property 'camp'
```



```
In [ ]:
```