# PYTHON ADVANCE ASSIGNMET BY SUDARSHAN PANDEY

April 18, 2024

[1]:

```
import numpy as np
#numpy is alias as np

import pandas as pd
#pandas is alias as pd

import matplotlib.pyplot as plt
#matplotlib is alias as plt

import seaborn as sns
#seaborn is alias as sns
```

[55]:

```
data = pd.read_csv("honeyproduction 1998-2021.csv")
print(data)
```

```
             State    numcol  yieldpercol   totalprod      stocks  priceperlb  \
0          Alabama   16000.0           71   1136000.0    159000.0        0.72
1          Arizona   55000.0           60   3300000.0   1485000.0        0.64
2         Arkansas   53000.0           65   3445000.0   1688000.0        0.59
3       California  450000.0           83  37350000.0  12326000.0        0.62
4         Colorado   27000.0           72   1944000.0   1594000.0        0.70
..             ...       ...          ...         ...         ...         ...
980       Virginia    6000.0           40    240000.0     79000.0        8.23
981     Washington   96000.0           32   3072000.0   1206000.0        2.52
982  West Virginia    6000.0           43    258000.0    136000.0        4.80
983      Wisconsin   42000.0           47   1974000.0    750000.0        2.81
984        Wyoming   38000.0           58   2204000.0    242000.0        2.07

      prodvalue  year
0      818000.0  1998
1     2112000.0  1998
2     2033000.0  1998
3    23157000.0  1998
4     1361000.0  1998
..          ...   ...
980   1975000.0  2021
981   7741000.0  2021
982   1238000.0  2021
```

```
983    5547000.0  2021
984    4562000.0  2021

[985 rows x 8 columns]
```

[56]:
```
data.head()
```

[56]:
```
        State    numcol  yieldpercol    totalprod      stocks  priceperlb  \
0     Alabama   16000.0           71    1136000.0    159000.0        0.72
1     Arizona   55000.0           60    3300000.0   1485000.0        0.64
2    Arkansas   53000.0           65    3445000.0   1688000.0        0.59
3  California  450000.0           83   37350000.0  12326000.0        0.62
4    Colorado   27000.0           72    1944000.0   1594000.0        0.70

    prodvalue  year
0    818000.0  1998
1   2112000.0  1998
2   2033000.0  1998
3  23157000.0  1998
4   1361000.0  1998
```

[57]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 985 entries, 0 to 984
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   State        985 non-null    object
 1   numcol       985 non-null    float64
 2   yieldpercol  985 non-null    int64
 3   totalprod    985 non-null    float64
 4   stocks       985 non-null    float64
 5   priceperlb   985 non-null    float64
 6   prodvalue    985 non-null    float64
 7   year         985 non-null    int64
dtypes: float64(5), int64(2), object(1)
memory usage: 61.7+ KB
```

[58]:
```
data.isnull().sum()
```

[58] :
```
State          0
numcol         0
yieldpercol    0
totalprod      0
stocks         0
priceperlb     0
```

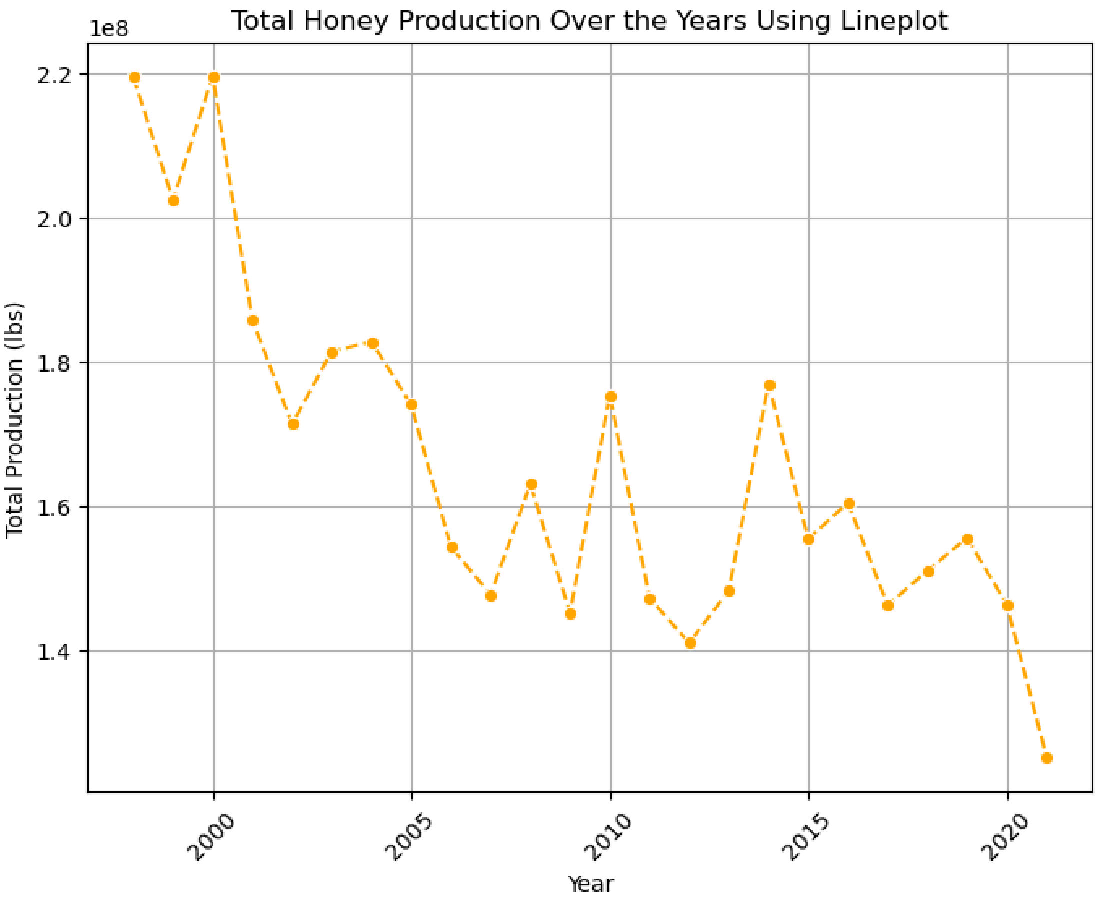
| | |
|---|---|
| prodvalue | 0 |

```
year          0
dtype: int64
```

## 1 Q1) How has honey production yield changed from 1998 to 2021?

[59]:

```python
# Group data by year and calculate total production for each year
total_yearly_production = data.groupby('year')['totalprod'].sum().reset_index()

# Plotting the trend over the years using a line plot
plt.figure(figsize=(8, 6))  #setting the plot size

#customising the line plot
sns.lineplot(data=total_yearly_production, x='year', y='totalprod', marker='o', color='orange', linestyle='dashed')

#setting the title of the plot
plt.title('Total Honey Production Over the Years Using Lineplot')

#setting the label of x-axis
plt.xlabel('Year')

#setting the label of y-axis
plt.ylabel('Total Production (lbs)')

#to plot the grid into the graph
plt.grid(True)


#setting this function to rotate the x-label
plt.xticks(rotation=45)

#to visualize the plot
plt.show()
```

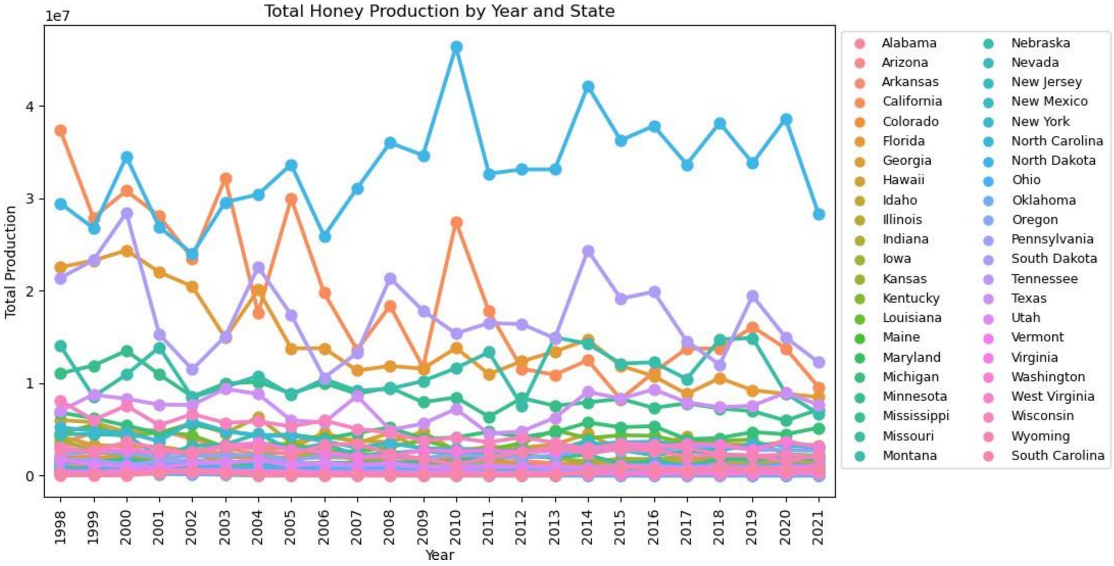


## 2 Q2) Over time, what are the major production trends across the states?

[60] :

```
#setting the plot size
plt.figure(figsize=(10,6))

#plotting the point plot
sns.pointplot(x='year', y='totalprod', data = data , estimator = sum , errorbar = None , hue='State')

#setting the title of the plot
plt.title('Total Honey Production by Year and State')

#customizing the legend
plt.legend(ncol = 2, fontsize=9, markerscale=0.8, bbox_to_anchor=(1,1))

# Customizing x-axis tick labels rotation
```

```
plt.xticks(rotation=90)

#setting the label of x-axis
plt.xlabel('Year')

#setting the label of y-axis
plt.ylabel('Total Production')

#to visualize the plot
plt.show()
```

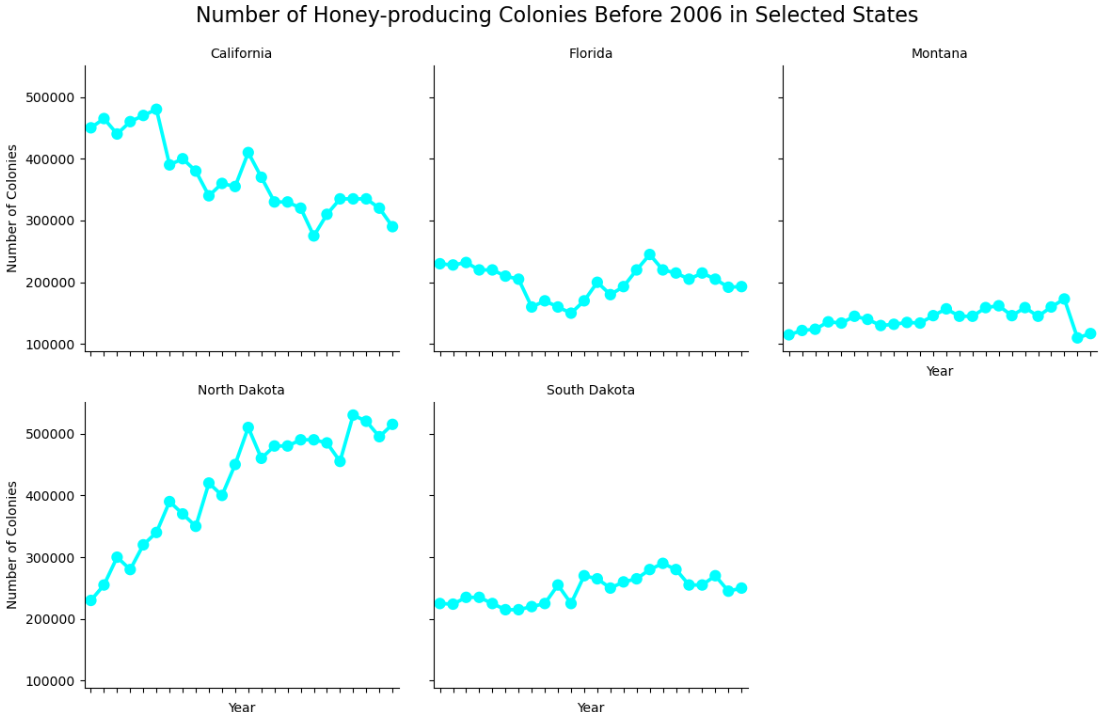


## Q3) Does the data show any trends in terms of the number of honey-producing colonies and yield per colony before 2006,which

## was when concern over Colony Collapse Disorder spread nation-wide?

[61] :

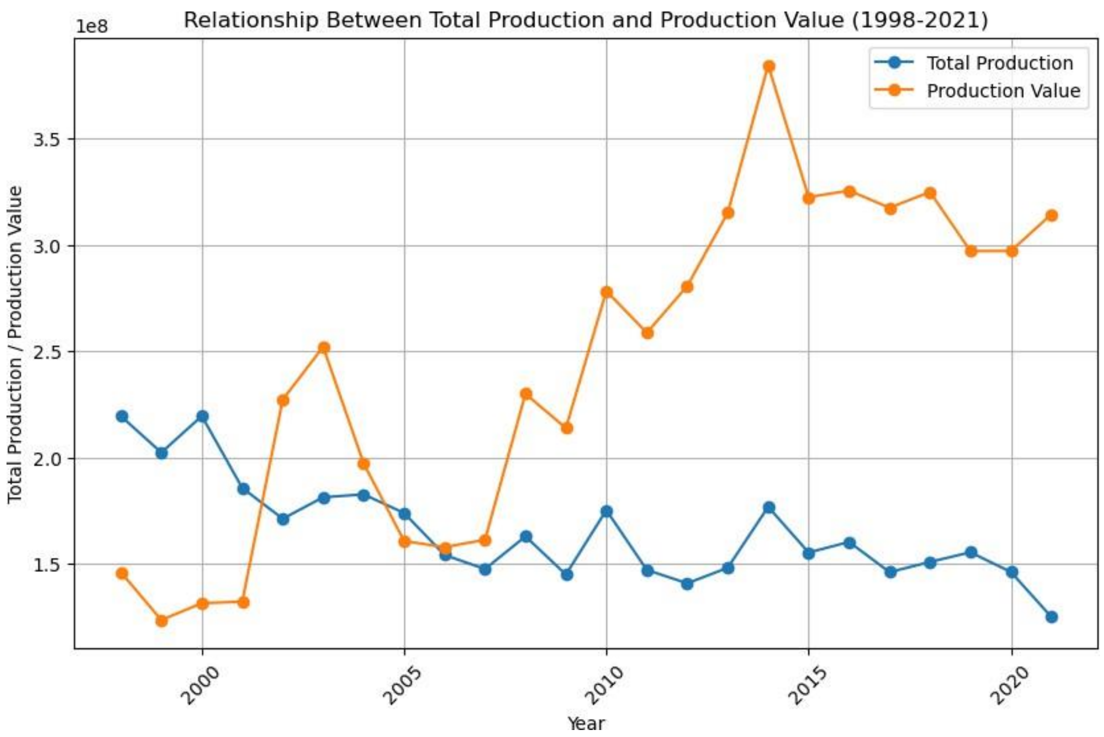
```
#setting the plot size
plt.figure(figsize=(4,4))

# Filter the data for specific states and before 2006
  cplot=sns.catplot(x='year',y='numcol',
                         data = data[data['State'].isin(['North Dakota',␣
  ↪'California','South  Dakota','Florida','Montana'])],
                              estimator=sum,col='State',kind='point',height=4,col_wrap=3, color='cyan'),

# Customizing x-axis tick labels rotation
cplot.set_xticklabels(rotation=90)

# Set axis labels
cplot.set_axis_labels("Year", "Number of Colonies")

# Set titles for each subplot based on state name
cplot.set_titles("{col_name}")

# Adjust spacing to make room for the main title
plt.subplots_adjust(top=0.9)

# Set main title for the entire figure
cplot.fig.suptitle('Number of Honey-producing Colonies Before 2006 in Selected States', fontsize=16)

#to visualize the plot
plt.show()
```



### Q4)Are there any patterns that can be observed between total honey production and the value of production every year?

[54]:

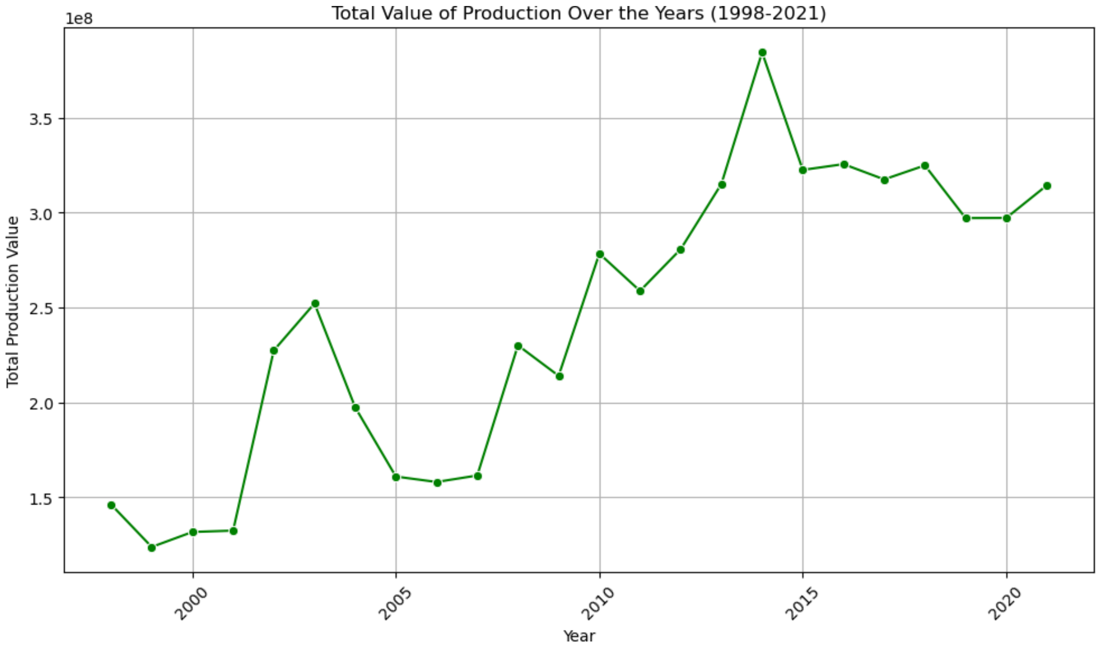
```python
# Group data by year and calculate total honey production and total value of production for each year
annual_production_data = data.groupby('year')[['totalprod', 'prodvalue']].sum().reset_index()

# Plotting the relationship using a line plot
plt.figure(figsize=(10, 6)) #setting plot size

# Plot total production
plt.plot(annual_production_data['year'], annual_production_data['totalprod'], label='Total Production', marker="o")

# Plot total production
plt.plot(annual_production_data['year'], annual_production_data['prodvalue'], label='Production Value', marker="o")

#setting the title of the plot
plt.title('Relationship Between Total Production and Production Value (1998-2021)')

#setting the label of x-axis
plt.xlabel('Year')

#setting the label of y-axis
plt.ylabel('Total Production / Production Value')

#setting the legend of the plot
plt.legend()

#setting grid
plt.grid(True)

#setting this function to rotate the x-label
plt.xticks(rotation=45)

#To visualize the plot
plt.show()
```



**Q5)How has the value of production, which in some sense could be tied to demand, changed every year?**

```
[24]: # Group data by year and calculate total value of production for each year
total_production_value_yearly = data.groupby('year')['prodvalue'].sum().
  ↪reset_index()

# Plotting the trend over the years using a line plot
plt.figure(figsize=(10, 6))#setting the plot size

#Customizing the line plot
sns.lineplot(data=total_production_value_yearly, x='year', y='prodvalue',␣
  ↪marker='o', color='green')

#setting the title of the plot
plt.title('Total Value of Production Over the Years (1998-2021)')

#setting the label of x-axis
plt.xlabel('Year')
```

```
#setting the lablel of y-axis
plt.ylabel('Total Production Value')

#to plot the grid
plt.grid(True)

#setting this function to rotate the x-label
plt.xticks(rotation=45)

#to visualize the plot
plt.show()
```


Total Value of Production Over the Years (1998-2021)
1e8
Total Production Value
Year