US Honey Hands on - 1

Tasks to Be Performed:

1. **Rarely Contributing States:** Identify the states that contribute to honey production less frequently.
2. **Top 5 Honey-Producing States:** Determine the top 5 states with the highest honey production.
3. **Change in Mean Average Price Over Time:** Analyze how the mean average price of honey has changed over the years.
4. **Year of Highest Production:** Find the year with the highest total honey production.
5. **State with the Highest Contribution in a Specific Year:** Identify the state with the highest honey production in a given year.
6. **States with the Highest Colonies in a Specific Year:** Determine the states with the highest number of bee colonies in a specific year.

```
# step 1


import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import os

os.chdir(r"C:\Users\harip\Desktop\intellipaat 24-09-2024\Python\Hands on\US honey combe
project")


# Verify the change

print("Current Directory:", os.getcwd())


import pandas as pd


df = pd.read_csv("US_honey_dataset_updated.csv")  # Make sure the file is inside this folder

print(df.head())
```

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
Current Directory: C:\Users\harip\Desktop\intellipaat 24-09-2024\Python\Hands on\US honey combe project
```

[21]:
```python
import pandas as pd

df = pd.read_csv("US_honey_dataset_updated.csv")  # Make sure the file is inside this folder
print(df.head())
```

```
   Unnamed: 0       state  colonies_number  yield_per_colony  production  \
0           0     Alabama            16000                58      928000
1           1     Arizona            52000                79     4108000
2           2    Arkansas            50000                60     3000000
3           3  California           420000                93    39060000
4           4    Colorado            45000                60     2700000

    stocks  average_price  value_of_production  year
0    28000           62.0               575000  1995
1   986000           68.0              2793000  1995
2   900000           64.0              1920000  1995
3  4687000           60.0             23436000  1995
4  1404000           68.0              1836000  1995
```

step 3 exploratory data analysis

df = df.drop(['Unnamed: 0'],axis = 1)

axis = 1 for drooping a column

df

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

[25]:
```python
df
```

[25]:

	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
0	Alabama	16000	58	928000	28000	62.00	575000	1995
1	Arizona	52000	79	4108000	986000	68.00	2793000	1995
2	Arkansas	50000	60	3000000	900000	64.00	1920000	1995
3	California	420000	93	39060000	4687000	60.00	23436000	1995
4	Colorado	45000	60	2700000	1404000	68.00	1836000	1995
...	...	...	...	...	...	...	...	...
1110	Virginia	6000	40	79000	79000	8.23	1975000	2021
1111	Washington	96000	32	1206000	1206000	2.52	7741000	2021
1112	WestVirginia	6000	43	136000	136000	4.80	1238000	2021
1113	Wisconsin	42000	47	750000	750000	2.81	5547000	2021
1114	Wyoming	38000	58	242000	242000	2.07	4562000	2021

1115 rows × 8 columns

[27]:
```python
# check the information about columns in our dataset
```

check the information about columns in our dataset

df.info()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

1115 rows × 8 columns

[27]:
```
# check the information about columns in our dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   state                1115 non-null   object
 1   colonies_number      1115 non-null   int64
 2   yield_per_colony     1115 non-null   int64
 3   production           1115 non-null   int64
 4   stocks               1115 non-null   int64
 5   average_price        1115 non-null   float64
 6   value_of_production  1115 non-null   int64
 7   year                 1115 non-null   int64
dtypes: float64(1), int64(6), object(1)
memory usage: 69.8+ KB
```

check how many unique states are there

df['state'].unique()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
 6   value_of_production  1115 non-null   int64
 7   year                 1115 non-null   int64
dtypes: float64(1), int64(6), object(1)
memory usage: 69.8+ KB
```

[30]:
```
# check how many unique states are there

df['state'].unique()
```

[30]:
```
array(['Alabama', 'Arizona', 'Arkansas', 'California', 'Colorado',
       'Florida', 'Georgia', 'Hawaii', 'Idaho', 'Illinois', 'Indiana',
       'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine', 'Maryland',
       'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',
       'Nebraska', 'Nevada', 'NewJersey', 'NewMexico', 'NewYork',
       'NorthCarolina', 'NorthDakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'SouthCarolina', 'SouthDakota', 'Tennessee',
       'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'WestVirginia', 'Wisconsin', 'Wyoming'], dtype=object)
```

step 3 exploratory data analysis

df = df.drop(['Unnamed: 0'],axis = 1)

axis = 1 for drooping a column

df

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

[23]:
```
df = df.drop(['Unnamed: 0'],axis = 1)

# axis = 1 for drooping a column
```

[25]:
```
df
```

[25]:

	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
0	Alabama	16000	58	928000	28000	62.00	575000	1995
1	Arizona	52000	79	4108000	986000	68.00	2793000	1995
2	Arkansas	50000	60	3000000	900000	64.00	1920000	1995
3	California	420000	93	39060000	4687000	60.00	23436000	1995
4	Colorado	45000	60	2700000	1404000	68.00	1836000	1995
...	...	...	...	...	...	...	...	...
1110	Virginia	6000	40	79000	79000	8.23	1975000	2021
1111	Washington	96000	32	1206000	1206000	2.52	7741000	2021
1112	WestVirginia	6000	43	136000	136000	4.80	1238000	2021
1113	Wisconsin	42000	47	750000	750000	2.81	5547000	2021
1114	Wyoming	38000	58	242000	242000	2.07	4562000	2021

1115 rows × 8 columns

check the information about columns in our dataset

df.info()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

1115 rows × 8 columns

[27]:
```
# check the information about columns in our dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   state                1115 non-null   object
 1   colonies_number      1115 non-null   int64
 2   yield_per_colony     1115 non-null   int64
 3   production           1115 non-null   int64
 4   stocks               1115 non-null   int64
 5   average_price        1115 non-null   float64
 6   value_of_production  1115 non-null   int64
 7   year                 1115 non-null   int64
dtypes: float64(1), int64(6), object(1)
memory usage: 69.8+ KB
```

check how many unique states are there

df['state'].unique()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
 6   value_of_production  1115 non-null   int64
 7   year                 1115 non-null   int64
dtypes: float64(1), int64(6), object(1)
memory usage: 69.8+ KB
```

[30]:
```
# check how many unique states are there

df['state'].unique()
```

[30]:
```
array(['Alabama', 'Arizona', 'Arkansas', 'California', 'Colorado',
       'Florida', 'Georgia', 'Hawaii', 'Idaho', 'Illinois', 'Indiana',
       'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine', 'Maryland',
       'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',
       'Nebraska', 'Nevada', 'NewJersey', 'NewMexico', 'NewYork',
       'NorthCarolina', 'NorthDakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'SouthCarolina', 'SouthDakota', 'Tennessee',
       'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'WestVirginia', 'Wisconsin', 'Wyoming'], dtype=object)
```

check how many unique states are there

df['state'].unique()

[32]:
```
# checking how many different years data I am having
df['year'].unique()

# 2000 - 1995 = 27
```

[32]:
```
array([1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
       2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
       2017, 2018, 2019, 2020, 2021], dtype=int64)
```

df['year'].nunique()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *


```
       'NorthCarolina', 'NorthDakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'SouthCarolina', 'SouthDakota', 'Tennessee',
       'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'WestVirginia', 'Wisconsin', 'Wyoming'], dtype=object)
```

[32]:
```
# checking how many different years data I am having
df['year'].unique()

# 2000 - 1995 = 27
```

[32]:
```
array([1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
       2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
       2017, 2018, 2019, 2020, 2021], dtype=int64)
```


[34]:
```
df['year'].nunique()
```

[34]:
```
27
```

data cleaning step

check for null values

df.isnull().sum()

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
2017, 2018, 2019, 2020, 2021], dtype=int64)
```

[34]:
```
df['year'].nunique()
```

[34]:
```
27
```

[36]:
```
# data cleaning step

# check for null values

df.isnull().sum()
```

[36]:
```
state                 0
colonies_number       0
yield_per_colony      0
production            0
stocks                0
average_price         0
value_of_production   0
year                  0
dtype: int64
```

check for the duplicate values

df.duplicated().sum()

no duplicate data in the dataset

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
stocks                0
average_price         0
value_of_production   0
year                  0
dtype: int64
```

[3]:
```
# check for the duplicate values

df.duplicated().sum()

# no duplicate data in the dataset
```

1) Which states are rarely contributing to honey production for the last 27 years?

data = df['state'].value_counts()

data

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
[40]: state
Alabama          27
Missouri         27
Arizona          27
NewJersey        27
NewYork          27
NorthCarolina    27
NorthDakota      27
Ohio             27
Oregon           27
Pennsylvania     27
SouthDakota      27
Tennessee        27
Texas            27
Utah             27
Vermont          27
Virginia         27
Washington       27
WestVirginia     27
Wisconsin        27
Montana          27
Nebraska         27
Mississippi      27
Minnesota        27
Arkansas         27
California       27
Colorado         27
Florida          27
Georgia          27
Hawaii           27
Idaho            27
Illinois         27
Indiana          27
```

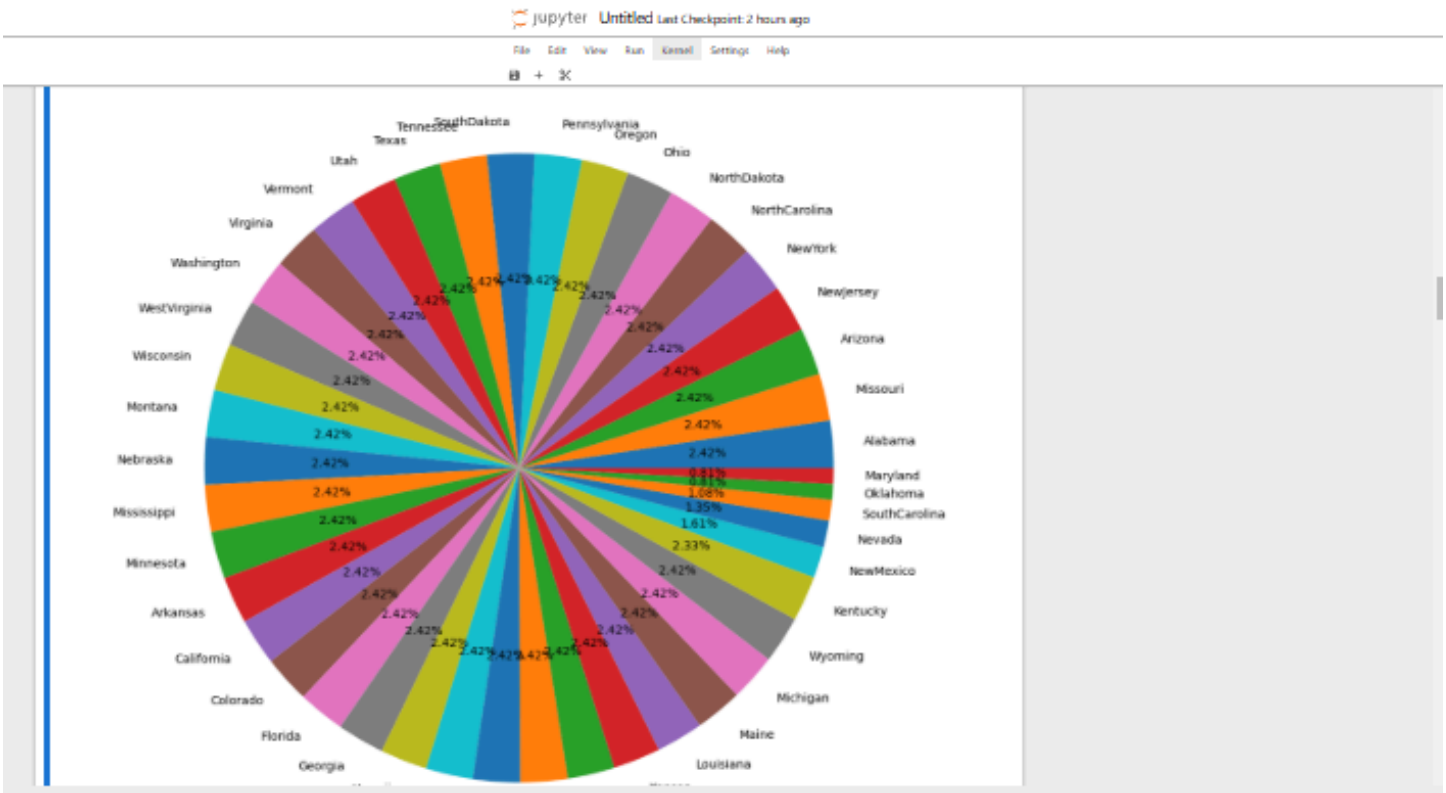
jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
Indiana          27
Iowa             27
Kansas           27
Louisiana        27
Maine            27
Michigan         27
Wyoming          27
Kentucky         26
NewMexico        18
Nevada           15
SouthCarolina    12
Oklahoma          9
Maryland          9
Name: count, dtype: int64
```

```
# draw a pie chart to compare the values


plt.figure(figsize = (12,12))


plt.pie(data.values,labels = data.index, autopct = "%0.2f%%" )

plt.show()
```



least honey producing states from the last 27 years

Maryland - 9

oklahoma - 9

2) Which are the top 5 Honey producing states in the US ?

```
plt.figure(figsize = (15,10))
```

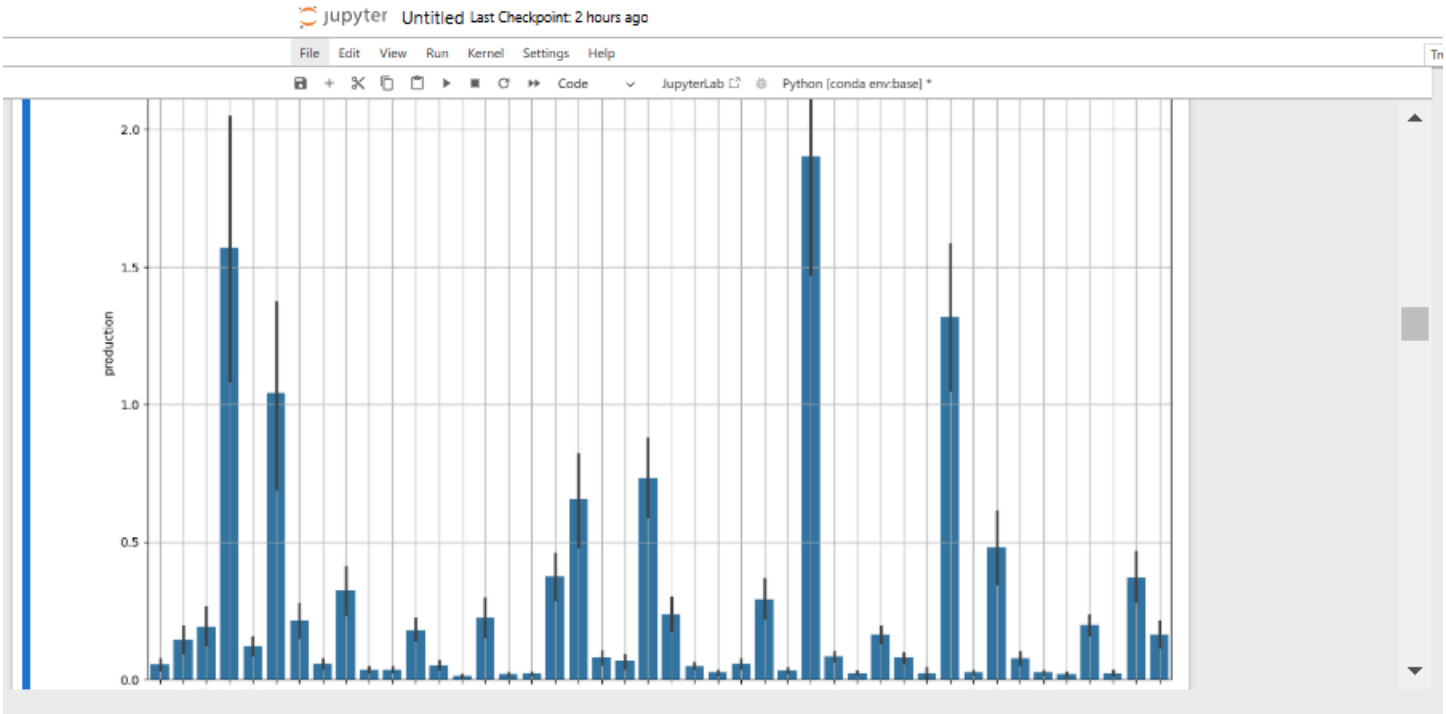
```
sns.barplot(x = df['state'],y= df['production'])
```

```
plt.xticks(rotation = 90)
```

```
plt.grid(True)
```

```
plt.title("state vs production")
```

```
plt.show()
```




```
# 1115 - entries

# 44- entries (unique)


# state -> production


# 1  arkansas - 87  1995


# 123 arkansas 56  2001



# 1. arkansas 87 + 56

# compare state by value of honey production


new_value = df.groupby('state').sum().reset_index()


new_value
```

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

```
# 1. arkansas 87 + 56
```

[52]:
```
# compare state by value of honey production

new_value = df.groupby('state').sum().reset_index()

new_value
```

[52]:

	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
0	Alabama	288000	1615	14467000	2987000	4057.61	28668000	54216
1	Arizona	854000	1490	38844000	17758000	2978.77	59772000	54216
2	Arkansas	908000	1886	51846000	20897000	2870.45	73781000	54216
3	California	10135000	1426	423876000	137611000	2954.06	653982000	54216
4	Colorado	799000	1520	32660000	18695000	3200.56	63844000	54216
5	Florida	5528000	1956	280934000	47037000	2967.52	509670000	54216
6	Georgia	1917000	1354	57426000	9932000	3260.68	145945000	54216
7	Hawaii	296000	2802	15420000	3508000	3719.13	49888000	54216
8	Idaho	2705000	1121	87188000	40503000	2869.71	139536000	54216
9	Illinois	236000	1562	9864000	4957000	5842.43	36782000	54216
10	Indiana	213000	1605	9686000	5286000	4183.16	24658000	54216
11	Iowa	962000	1659	48607000	29351000	3341.75	87563000	54216

sort the values in descending order

new_df = new_value.sort_values(by = 'production', ascending = False)

new_df

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

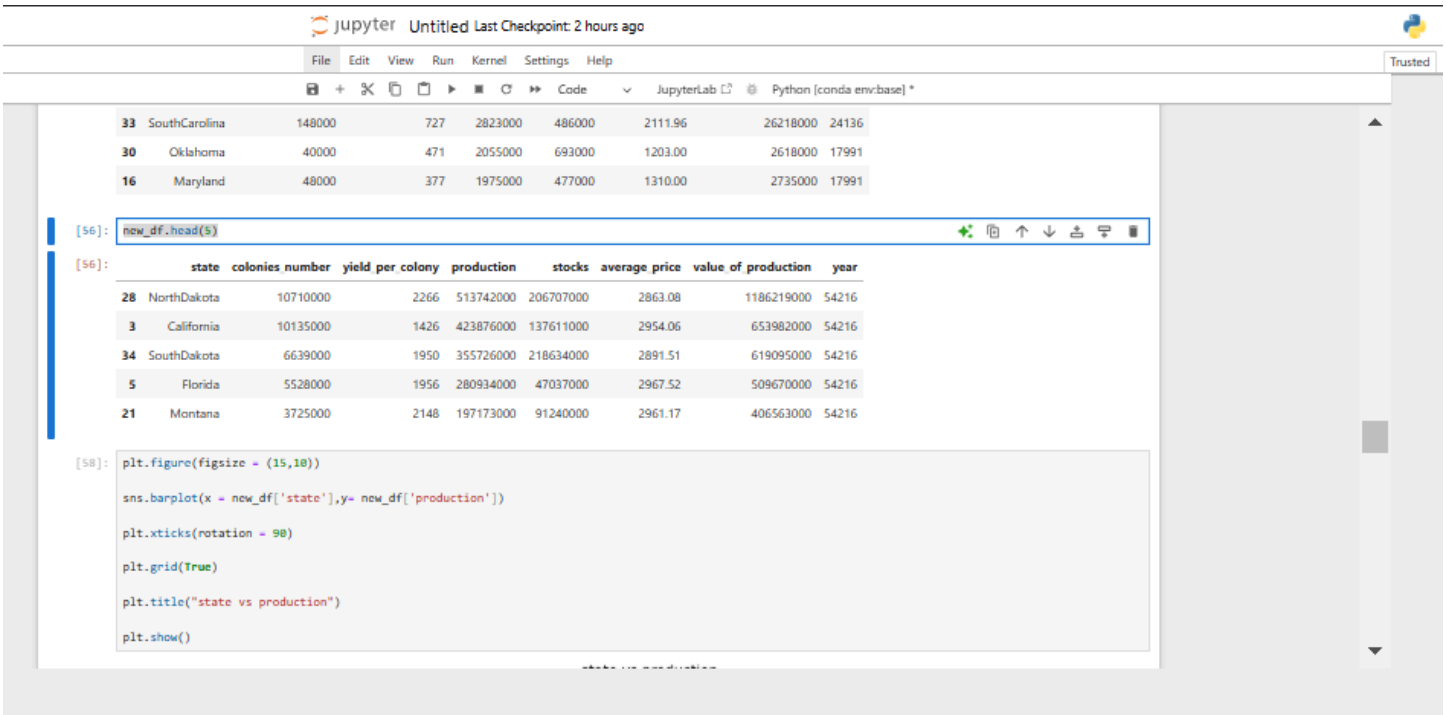
Code JupyterLab Python [conda env:base] *

[54]:
```
# sort the values in descending order

new_df = new_value.sort_values(by = 'production', ascending = False)

new_df
```

[54]:

	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
28	NorthDakota	10710000	2266	513742000	206707000	2863.08	1186219000	54216
3	California	10135000	1426	423876000	137611000	2954.06	653982000	54216
34	SouthDakota	6639000	1950	355726000	218634000	2891.51	619095000	54216
5	Florida	5528000	1956	280934000	47037000	2967.52	509670000	54216
21	Montana	3725000	2148	197173000	91240000	2961.17	406563000	54216
18	Minnesota	3498000	1885	176581000	51908000	2887.89	310462000	54216
36	Texas	2876000	1921	129441000	41022000	2965.74	284720000	54216
17	Michigan	2168000	1753	101063000	58401000	3311.97	217841000	54216
42	Wisconsin	1698000	1896	99909000	59166000	3410.87	176442000	54216
8	Idaho	2705000	1121	87188000	40503000	2869.71	139536000	54216
26	NewYork	1588000	1745	78444000	42535000	3663.52	177921000	54216
22	Nebraska	1279000	1729	63655000	38937000	3004.31	106998000	54216
14	Louisiana	1066000	2506	60568000	12688000	2783.02	129601000	54216
6	Georgia	1917000	1354	57426000	9932000	3260.68	145945000	54216

new_df.head(5)



plt.figure(figsize = (15,10))

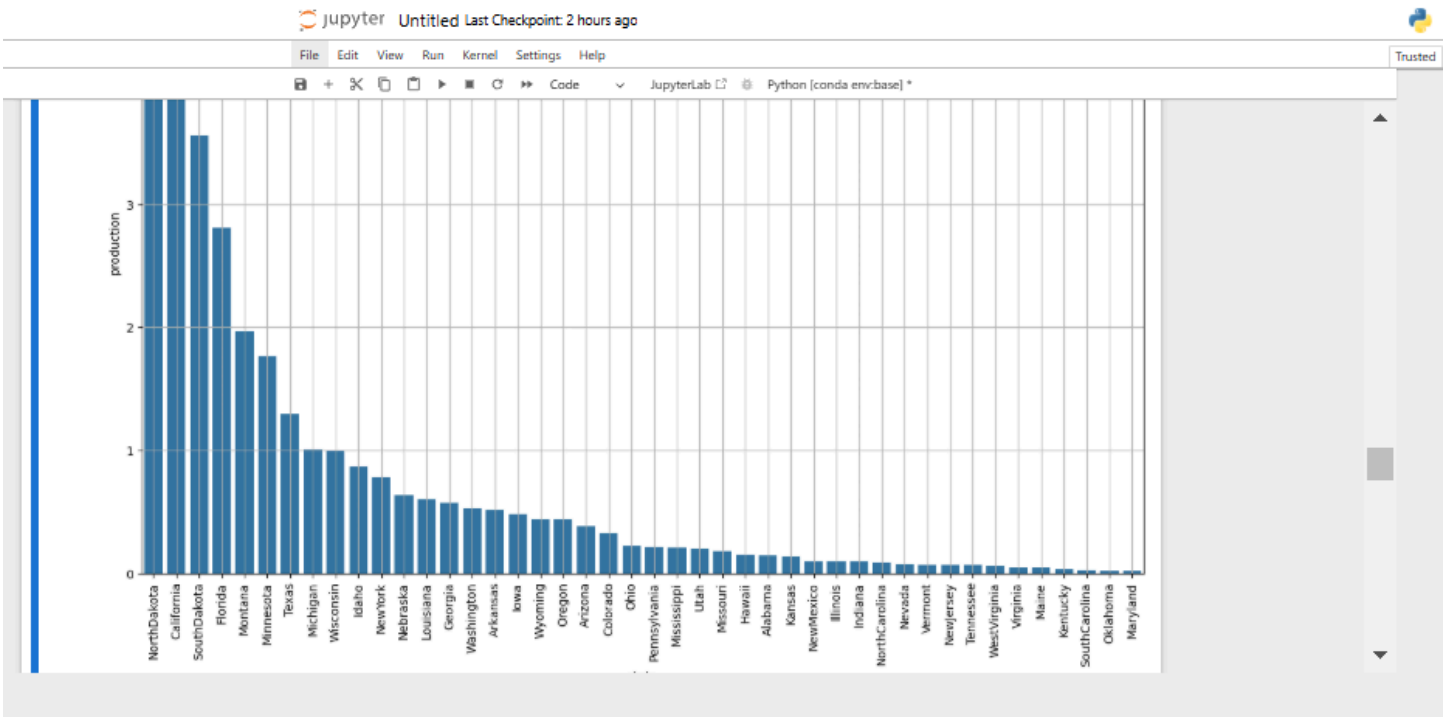
sns.barplot(x = new_df['state'],y= new_df['production'])

plt.xticks(rotation = 90)

plt.grid(True)

plt.title("state vs production")

plt.show()



top 5 honey producing states

1 - north dakota

2 - california

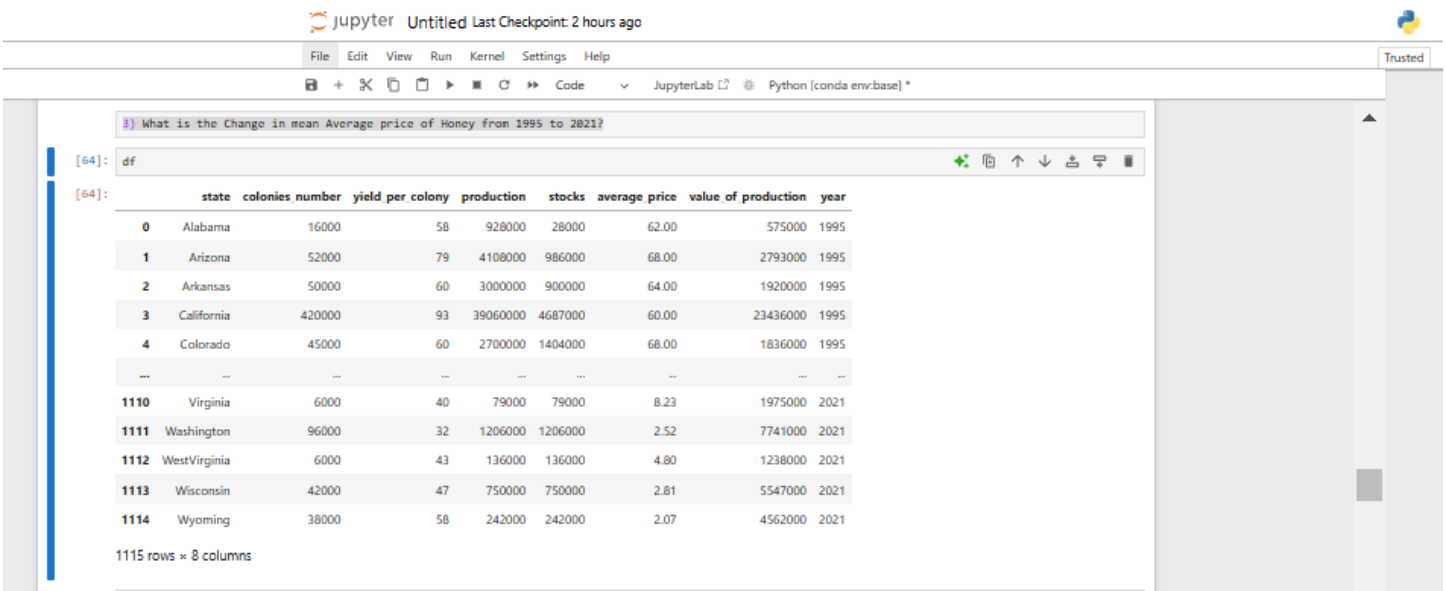
3- south dakota

4 - florida

5 - montana

3) What is the Change in mean Average price of Honey from 1995 to 2021?

df



	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
0	Alabama	16000	58	928000	28000	62.00	575000	1995
1	Arizona	52000	79	4108000	986000	68.00	2793000	1995
2	Arkansas	50000	60	3000000	900000	64.00	1920000	1995
3	California	420000	93	39060000	4687000	60.00	23436000	1995
4	Colorado	45000	60	2700000	1404000	68.00	1836000	1995
...	...	...	...	...	...	...	...	...
1110	Virginia	6000	40	79000	79000	8.23	1975000	2021
1111	Washington	96000	32	1206000	1206000	2.52	7741000	2021
1112	WestVirginia	6000	43	136000	136000	4.80	1238000	2021
1113	Wisconsin	42000	47	750000	750000	2.81	5547000	2021
1114	Wyoming	38000	58	242000	242000	2.07	4562000	2021

1115 rows × 8 columns

```
df2 = df.groupby('year').mean(['average_price']).reset_index()
```

```
df2
```

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *

[68]:

	year	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production
0	1995	59977.272727	66.909091	4.778909e+06	9.575000e+05	74.840909	3.121000e+06
1	1996	58181.818182	70.068182	4.499886e+06	1.066455e+06	99.568182	4.085773e+06
2	1997	59651.162791	68.953488	4.445953e+06	1.601256e+06	91.325581	3.386000e+06
3	1998	60883.720930	69.953488	5.100488e+06	1.871488e+06	83.720930	3.395302e+06
4	1999	62186.046512	65.465116	4.757791e+06	1.839698e+06	80.325581	2.888070e+06
5	2000	60860.465116	67.534884	5.123721e+06	1.997395e+06	79.023256	3.047023e+06
6	2001	58139.534884	65.209302	4.311698e+06	1.501791e+06	88.465116	2.936302e+06
7	2002	57181.818182	67.272727	3.880273e+06	8.831591e+05	133.204545	5.016977e+06
8	2003	58681.818182	62.522727	4.107750e+06	9.220227e+05	151.068182	5.791659e+06
9	2004	63325.000000	65.025000	4.559475e+06	1.523100e+06	132.350000	4.976100e+06
10	2005	58341.463415	64.268293	4.240415e+06	1.515732e+06	116.341463	3.795390e+06
11	2006	57926.829268	61.853659	3.759024e+06	1.469610e+06	134.341463	3.886902e+06
12	2007	59121.951220	59.390244	3.600220e+06	1.273488e+06	140.170732	3.801634e+06
13	2008	55756.097561	61.000000	3.904780e+06	1.227195e+06	160.878049	5.466171e+06
14	2009	59682.926829	53.780488	3.496098e+06	9.030732e+05	176.195122	5.063463e+06
15	2010	66650.000000	56.275000	1.119925e+06	1.119925e+06	192.900000	6.959250e+06
16	2011	61650.000000	54.775000	9.137500e+05	9.137500e+05	216.725000	6.467200e+06
17	2012	62725.000000	55.175000	7.915500e+05	7.915500e+05	236.700000	7.018125e+06

```
plt.figure(figsize = (15,10))
```

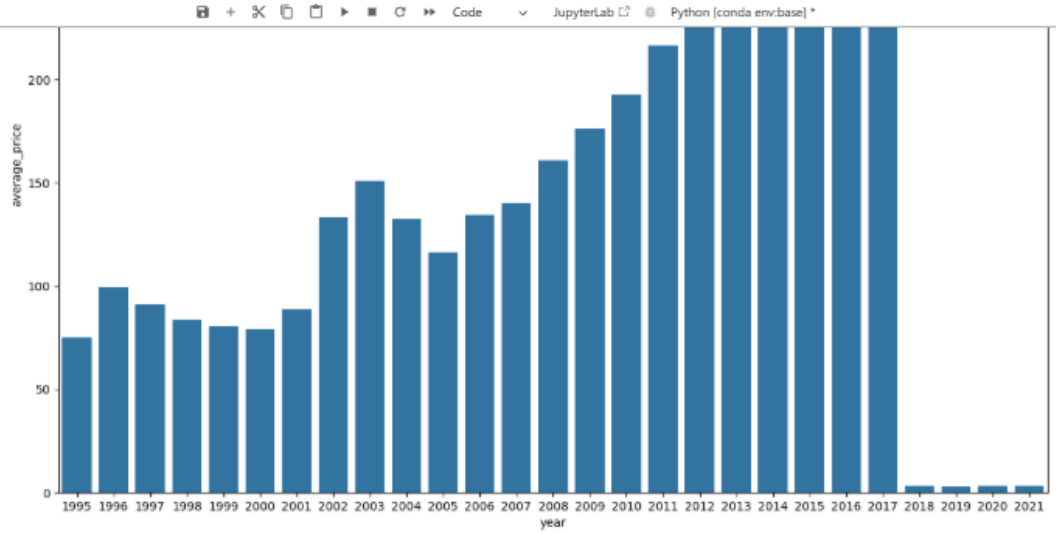
```
sns.barplot(x = df2['year'] , y = df2['average_price'])
```

```
plt.show()
```

jupyter Untitled Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python [conda env:base] *



[72]:
```
# average price was highest in the year -2017
# average price was lowest in the year - 2019
```

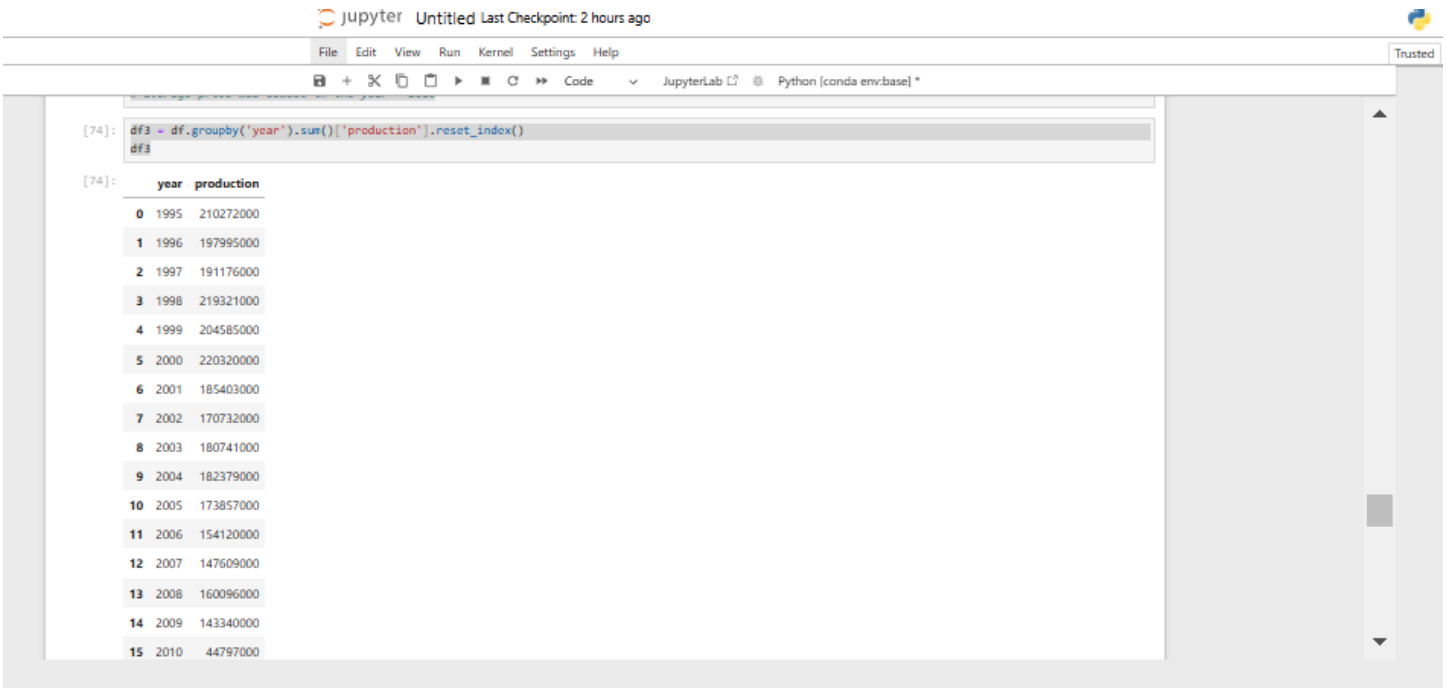
```
# average price was highest in the year -2017
```

average price was lowest in the year - 2019

4) Which was the year when production of Honey in wholeUS was the highest?

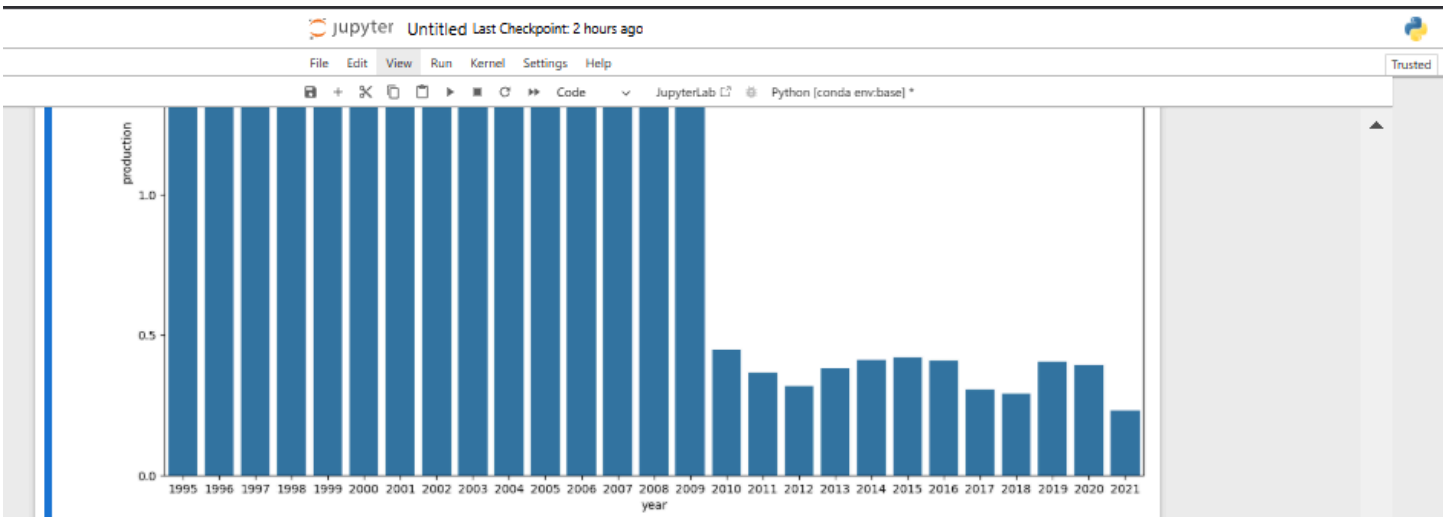
```
df3 = df.groupby('year').sum()['production'].reset_index()
```

```
df3
```



	year	production
0	1995	210272000
1	1996	197995000
2	1997	191176000
3	1998	219321000
4	1999	204585000
5	2000	220320000
6	2001	185403000
7	2002	170732000
8	2003	180741000
9	2004	182379000
10	2005	173857000
11	2006	154120000
12	2007	147609000
13	2008	160096000
14	2009	143340000
15	2010	44797000

```
plt.figure(figsize = (15,10))

sns.barplot(x = df3['year'],y = df3['production'])

plt.show()
```



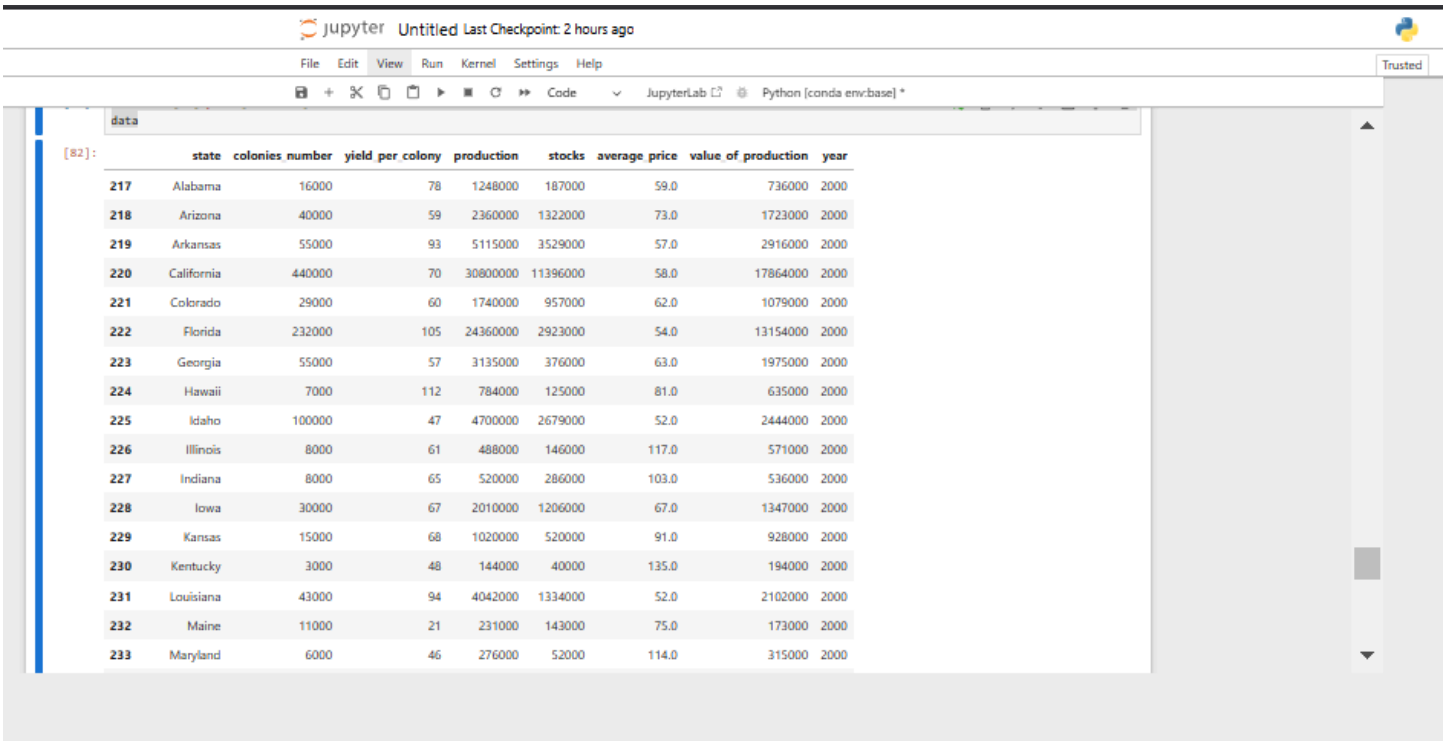
conclusion

The year 2000 the production was highest in USA

5) From the above inference we get the production was highest in the year 2000, now let infer which state was having highest contribution in that year

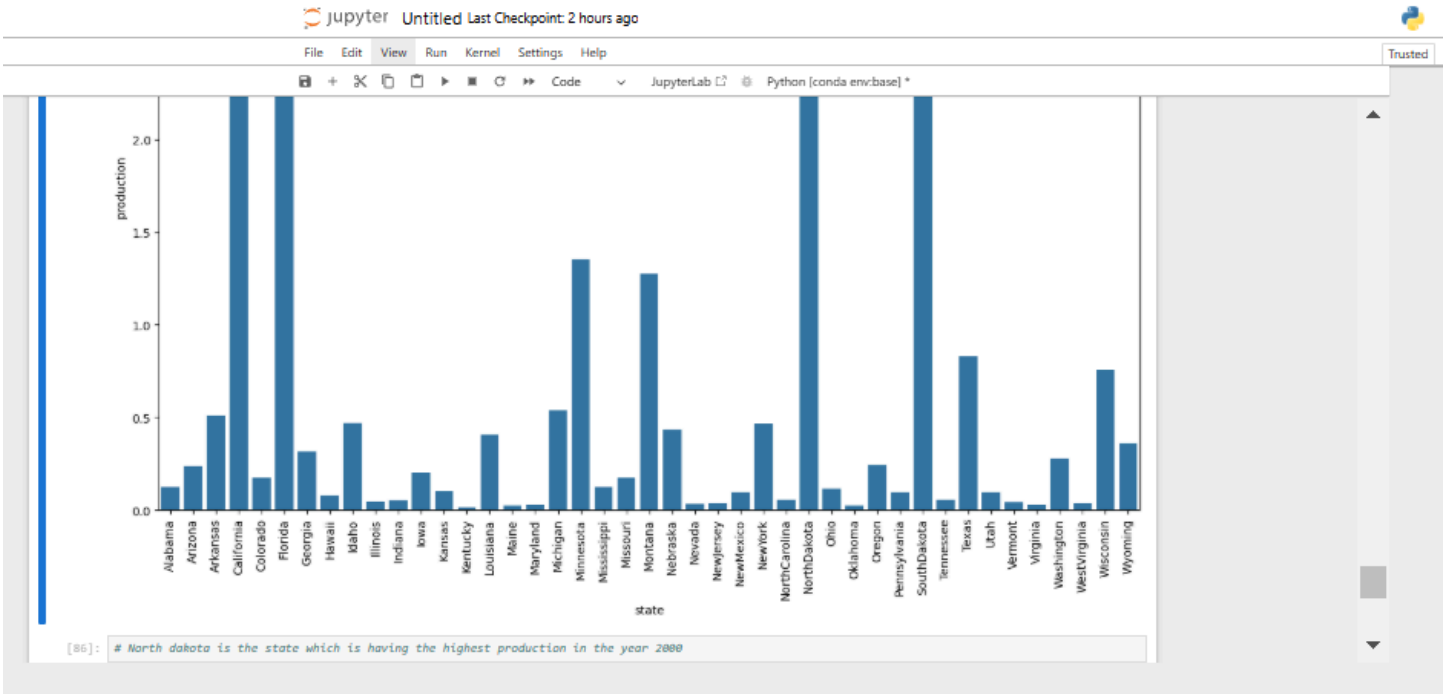
```
data = df[df['year'] == 2000]
```

```
data
```



	state	colonies_number	yield_per_colony	production	stocks	average_price	value_of_production	year
217	Alabama	16000	78	1248000	187000	59.0	736000	2000
218	Arizona	40000	59	2360000	1322000	73.0	1723000	2000
219	Arkansas	55000	93	5115000	3529000	57.0	2916000	2000
220	California	440000	70	30800000	11396000	58.0	17864000	2000
221	Colorado	29000	60	1740000	957000	62.0	1079000	2000
222	Florida	232000	105	24360000	2923000	54.0	13154000	2000
223	Georgia	55000	57	3135000	376000	63.0	1975000	2000
224	Hawaii	7000	112	784000	125000	81.0	635000	2000
225	Idaho	100000	47	4700000	2679000	52.0	2444000	2000
226	Illinois	8000	61	488000	146000	117.0	571000	2000
227	Indiana	8000	65	520000	286000	103.0	536000	2000
228	Iowa	30000	67	2010000	1206000	67.0	1347000	2000
229	Kansas	15000	68	1020000	520000	91.0	928000	2000
230	Kentucky	3000	48	144000	40000	135.0	194000	2000
231	Louisiana	43000	94	4042000	1334000	52.0	2102000	2000
232	Maine	11000	21	231000	143000	75.0	173000	2000
233	Maryland	6000	46	276000	52000	114.0	315000	2000

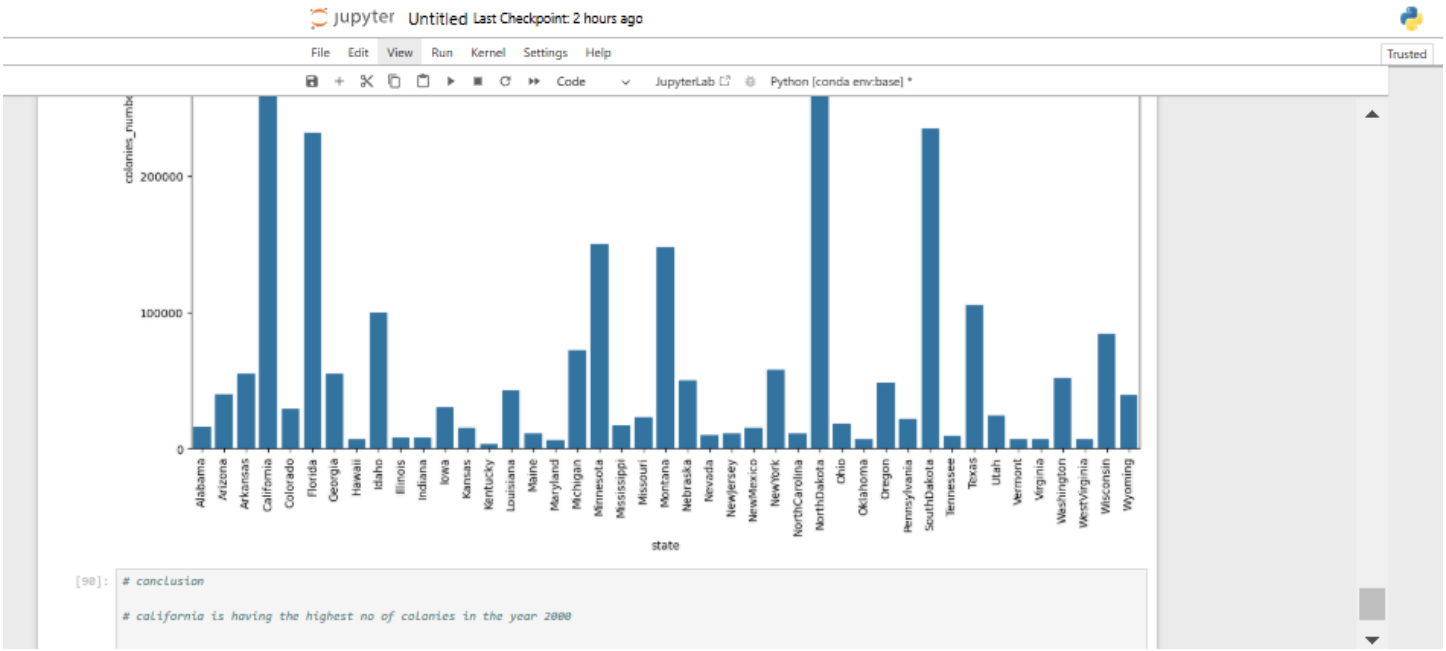
```
plt.figure(figsize = (15,10))

sns.barplot(x = data['state'],y = data['production'])

plt.xticks(rotation = 90)

plt.show()
```



North dakota is the state which is having the highest production in the year 2000

6) Which states have the highest no. of colonies in the year 2000?

```
plt.figure(figsize = (15,10))

sns.barplot(x = data['state'],y = data['colonies_number'])

plt.xticks(rotation = 90)

plt.show()
```



conclusion

california is having the highest no of colonies in the year 2000