ananyatiwari14 / J233-Data-Project_Ananya-Tiwari Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

J233-Data-Project_Ananya-Tiwari / j33 finals.ipynb

ananyatiwari14 switching to jupyter lab History

1 contributor

2.14 MB

In [1]:
```
import pandas as pd
import numpy as np
import altair as alt
import matplotlib as plt
```

In [2]:
```
alt.renderers.enable('mimetype')
```

Out[2]:
```
RendererRegistry.enable('mimetype')
```

In [3]:
```
df = pd.read_csv('/Users/ananyatiwari/Desktop/India Agriculture Crop Production.csv')
```

In [4]:
```
df.to_csv('Agricultural production since 1997 in India') ## To
```

After discussing the other datasets with you during office hours, I realized that I should try to find data from more legitimate sources. Unfortunately, I was unsuccessful in reaching the US Foreign Agricultural Service for data. However, I found a great Indian government data website where I was able to find a great dataset which was similar to what I found on Kaggle.

This dataset is available here: https://aps.dac.gov.in/APY/Public_Report1.aspx

I had to go on the Ministry of Agriculture and Farmers Welfare website and in particular, here: https://aps.dac.gov.in/Home.aspx?ReturnUrl=%2f where I chose APY. This led me to this page - https://aps.dac.gov.in/APY/Index.htm where I chose District wise crop production statistics. And then I could generate my report from these selections - https://aps.dac.gov.in/APY/Public_Report1.aspx . I chose to check all the boxes in every field.

It was downloaded as an Excel file.

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344208 entries, 0 to 344207
Data columns (total 11 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Unnamed: 0        344208 non-null  int64  
 1   State             344208 non-null  object 
 2   District          344208 non-null  object 
 3   Crop              344099 non-null  object 
 4   Year              344208 non-null  object 
 5   Season            344208 non-null  object 
 6   Area              344099 non-null  float64
 7   Area Units        344208 non-null  object 
 8   Production        339187 non-null  float64
 9   Production Units  344208 non-null  object 
 10  Yield             344099 non-null  float64
dtypes: float64(3), int64(1), object(7)
memory usage: 28.9+ MB
```

In [6]:
```
del df['Unnamed: 0']
```

In [7]:
```
df.head(10)
```

Out[7]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
0	Andaman and Nicobar Islands	NICOBARS	Arecanut	2001-02	Kharif	1254.0	Hectare	2061.0	Tonnes	1.643541
	Andaman									

1	and Nicobar Islands	NICOBARS	Arecanut	2002-03	Whole Year	1258.0	Hectare	2083.0	Tonnes	1.655803
2	Andaman and Nicobar Islands	NICOBARS	Arecanut	2003-04	Whole Year	1261.0	Hectare	1525.0	Tonnes	1.209358
3	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001-02	Kharif	3100.0	Hectare	5239.0	Tonnes	1.690000
4	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002-03	Whole Year	3105.0	Hectare	5267.0	Tonnes	1.696296
5	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2003-04	Whole Year	3118.0	Hectare	5182.0	Tonnes	1.661963
6	Andaman and Nicobar Islands	NICOBARS	Banana	2002-03	Whole Year	213.0	Hectare	1278.0	Tonnes	6.000000
7	Andaman and Nicobar Islands	NICOBARS	Banana	2003-04	Whole Year	266.0	Hectare	1763.0	Tonnes	6.627820
8	Andaman and Nicobar Islands	SOUTH ANDAMANS	Banana	2002-03	Whole Year	1524.0	Hectare	10882.0	Tonnes	7.140420
9	Andaman and Nicobar Islands	SOUTH ANDAMANS	Banana	2003-04	Whole Year	1530.0	Hectare	11558.0	Tonnes	7.554248

This dataset has detailed information on the various districts in each state and union territory, and the crop production for various seasons. Rabi is the winter season, and kharif is the summer season. Rabi crops are sown in the winter around November, and harvested in the spring. Kharif crops are sown in the summer around May and harvested in autumn in around October/November. For my purposes, I will choose to stick to these two seasons as they are the main cropping seasons.

Since I am interested in how the crop production of various yields has progressed over time, I will choose specific states-districts for my analysis. This is because in India certain areas are more vulnerable to climatic stresses or unpredictable weather than other places. I am still doing my research on which areas are more prone to such changes. But, I will use this dataset to see if there are some irregular patterns in certain crops/fruits too. It will be a repetitive process of selecting various kinds of datasets and creating line charts to see the trends over the decades from 1997-2020.

I noticed that the year column is formatted in YYYY-YY and I wish to make it simpler, into a YYYY format. This will help later in the data vizualizations. I will convert the column to string dtype, and strip it.

In [8]:

```
df["Year"]=df["Year"].astype('string')
print(df.dtypes)
```

```
State                 object
District              object
Crop                  object
Year                  string
Season                object
Area                 float64
Area Units            object
Production           float64
Production Units      object
Yield                float64
dtype: object
```

In [9]:

```
df['Year'] = df['Year'].str[:4]
df.head(10)
```

Out[9]:

	State	District	Crop	Year	Season	Area	Area	Production	Production	Yield

							Units		Units	
0	Andaman and Nicobar Islands	NICOBARS	Arecanut	2001	Kharif	1254.0	Hectare	2061.0	Tonnes	1.643541
1	Andaman and Nicobar Islands	NICOBARS	Arecanut	2002	Whole Year	1258.0	Hectare	2083.0	Tonnes	1.655803
2	Andaman and Nicobar Islands	NICOBARS	Arecanut	2003	Whole Year	1261.0	Hectare	1525.0	Tonnes	1.209358
3	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001	Kharif	3100.0	Hectare	5239.0	Tonnes	1.690000
4	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002	Whole Year	3105.0	Hectare	5267.0	Tonnes	1.696296
5	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2003	Whole Year	3118.0	Hectare	5182.0	Tonnes	1.661963
6	Andaman and Nicobar Islands	NICOBARS	Banana	2002	Whole Year	213.0	Hectare	1278.0	Tonnes	6.000000
7	Andaman and Nicobar Islands	NICOBARS	Banana	2003	Whole Year	266.0	Hectare	1763.0	Tonnes	6.627820
8	Andaman and Nicobar Islands	SOUTH ANDAMANS	Banana	2002	Whole Year	1524.0	Hectare	10882.0	Tonnes	7.140420
9	Andaman and Nicobar Islands	SOUTH ANDAMANS	Banana	2003	Whole Year	1530.0	Hectare	11558.0	Tonnes	7.554248

In [10]:

```
df["Year"]=df["Year"].astype('int')
print(df.dtypes)
```

```
State                 object
District              object
Crop                  object
Year                   int64
Season                object
Area                 float64
Area Units            object
Production           float64
Production Units      object
Yield                float64
dtype: object
```

In [11]:

```
df.head(5)
```

Out[11]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
0	Andaman and Nicobar Islands	NICOBARS	Arecanut	2001	Kharif	1254.0	Hectare	2061.0	Tonnes	1.643541
1	Andaman and Nicobar Islands	NICOBARS	Arecanut	2002	Whole Year	1258.0	Hectare	2083.0	Tonnes	1.655803
2	Andaman and Nicobar Islands	NICOBARS	Arecanut	2003	Whole Year	1261.0	Hectare	1525.0	Tonnes	1.209358
3	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001	Kharif	3100.0	Hectare	5239.0	Tonnes	1.690000
	Andaman									

4	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002	Whole Year	3105.0	Hectare	5267.0	Tonnes	1.696296

In [12]:
```
df.Crop.unique()
```

Out[12]:
```
array(['Arecanut', 'Banana', 'Black pepper', 'Cashewnut', 'Coconut',
       'Dry chillies', 'Ginger', 'Other Kharif pulses', 'other oilseeds',
       'Rice', 'Sugarcane', 'Sweet potato', 'Arhar/Tur', 'Bajra',
       'Castor seed', 'Coriander', 'Cotton(lint)', 'Gram', 'Groundnut',
       'Horse-gram', 'Jowar', 'Linseed', 'Maize', 'Mesta',
       'Moong(Green Gram)', 'Niger seed', 'Onion', 'Other Rabi pulses',
       'Potato', 'Ragi', 'Rapeseed &Mustard', 'Safflower', 'Sesamum',
       'Small millets', 'Soyabean', 'Sunflower', 'Tapioca', 'Tobacco',
       'Turmeric', 'Urad', 'Wheat', 'Oilseeds total', 'Jute', 'Masoor',
       'Peas & beans (Pulses)', 'Barley', 'Garlic', 'Khesari', 'Sannhamp',
       'Guar seed', 'Moth', 'Cardamom', 'Other Cereals', 'Cowpea(Lobia)',
       'Dry Ginger', 'Other Summer Pulses', nan], dtype=object)
```

Right now, I am exploring the dataset so that I can see which crops, states, districts, etc, are in it. Above, is a list of the crops whose production values across different states is being calculated.

In [13]:
```
df.Season.unique()
```

Out[13]:
```
array(['Kharif', 'Whole Year', 'Rabi', 'Autumn', 'Summer', 'Winter'],
      dtype=object)
```

Here I am seeing six seasons, and I know that many of them overlap with each other. Kharif season runs from May-September/October, and the Rabi season begins from November to April/May. These two seasons tend to incorporate the Summer and Winter seasons. I will use these two seasons for my analysis.

When it comes to the states, I would need to know which ones since I want to explore this dataset state-wise, and not crop-wise, for more precise patterns.

In [14]:
```
df.State.unique()
```

Out[14]:
```
array(['Andaman and Nicobar Islands', 'Andhra Pradesh',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
       'Chhattisgarh', 'Dadra and Nagar Haveli', 'Daman and Diu', 'Delhi',
       'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh',
       'Jammu and Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',
       'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',
       'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan',
       'Sikkim', 'Tamil Nadu', 'Tripura', 'Uttar Pradesh', 'Uttarakhand',
       'West Bengal', 'Telangana'], dtype=object)
```

Each state has its districts, which are usually but not always, predominated by one or two crop types. For example, West Bengal is known for being a major cultivator of rice. However, rice is grown all over India also, such as in the states of Punjab, and others.In my exploration of this dataset (which I discussed with you during office hours) I had already found a few crops showing an alarming rate of decline, and I was later able to find news sources to help me figure out why this might be the case. Production of some crops have increased, though, and these variations are very crop and region specific. I will explore an interesting case of Punjab, and for this, I will subset data from the state, particularly its rice production.

Production for all states, all crops, by year

In [15]:
```
production_all = df.groupby(['State', 'Crop', 'Year', 'Season']).sum()[['Production']].reset
production_all
```

Out[15]:

	State	Crop	Year	Season	Production
0	Andaman and Nicobar Islands	Arecanut	2000	Kharif	7200.00

1	Andaman and Nicobar Islands	Arecanut	2001	Kharif	7300.00
2	Andaman and Nicobar Islands	Arecanut	2002	Whole Year	7350.00
3	Andaman and Nicobar Islands	Arecanut	2003	Whole Year	6707.00
4	Andaman and Nicobar Islands	Arecanut	2004	Whole Year	4781.05
...	...	...	...	...	...
21383	West Bengal	Wheat	2015	Rabi	788503.00
21384	West Bengal	Wheat	2016	Rabi	862712.00
21385	West Bengal	Wheat	2017	Rabi	362744.00
21386	West Bengal	Wheat	2018	Rabi	337751.00
21387	West Bengal	Wheat	2019	Rabi	509970.00

21388 rows × 5 columns

In [16]:

```
# subset for Punjab and Rice
production_all[
    (production_all['State'] == 'Punjab') &
    (production_all['Crop'] == 'Rice')
].reset_index()
```

Out[16]:

	index	State	Crop	Year	Season	Production
0	15849	Punjab	Rice	1997	Kharif	7904000.0
1	15850	Punjab	Rice	1998	Kharif	7940000.0
2	15851	Punjab	Rice	1999	Kharif	8716000.0
3	15852	Punjab	Rice	2000	Kharif	9154000.0
4	15853	Punjab	Rice	2001	Kharif	8816000.0
5	15854	Punjab	Rice	2002	Kharif	8880000.0
6	15855	Punjab	Rice	2003	Kharif	9656000.0
7	15856	Punjab	Rice	2004	Kharif	10437000.0
8	15857	Punjab	Rice	2005	Kharif	10193000.0
9	15858	Punjab	Rice	2006	Kharif	10138000.0
10	15859	Punjab	Rice	2007	Kharif	10489000.0
11	15860	Punjab	Rice	2008	Kharif	11000000.0
12	15861	Punjab	Rice	2009	Kharif	11236000.0
13	15862	Punjab	Rice	2010	Kharif	10837000.0
14	15863	Punjab	Rice	2011	Kharif	10542000.0
15	15864	Punjab	Rice	2012	Kharif	11390000.0
16	15865	Punjab	Rice	2013	Kharif	11267000.0
17	15866	Punjab	Rice	2014	Kharif	11107000.0
18	15867	Punjab	Rice	2015	Kharif	11823000.0
19	15868	Punjab	Rice	2016	Kharif	12638000.0
20	15869	Punjab	Rice	2017	Kharif	13382000.0
21	15870	Punjab	Rice	2018	Kharif	12822000.0
22	15871	Punjab	Rice	2019	Kharif	12675000.0

Crop production in Punjab

HYPOTHESIS: One interesting case study of sorts is Punjab, where rice yeilds have declined over time, due to various reasons. In this article, Punjab's ground-water level has been declining sharply ever since rice

cultivation was introduced into the state. Rice is a water-intensive crop. The government there is pushing the farmers to diversify the crops growns to reduce rice cultivation to help with this issue. This could be a good reason why over time rice cultivation in the area is declining. This policy is also a few years old only.

In [17]:

```
df_pj  =  df[(df['State'] == 'Punjab')].reset_index(drop=True)
df_pj
```

Out[17]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
0	Punjab	AMRITSAR	Arhar/Tur	2001	Kharif	1400.0	Hectare	1100.0	Tonnes	0.785714
1	Punjab	AMRITSAR	Arhar/Tur	2002	Kharif	1200.0	Hectare	1000.0	Tonnes	0.833333
2	Punjab	AMRITSAR	Arhar/Tur	2003	Kharif	1500.0	Hectare	1400.0	Tonnes	0.933333
3	Punjab	BATHINDA	Arhar/Tur	2003	Kharif	100.0	Hectare	100.0	Tonnes	1.000000
4	Punjab	FARIDKOT	Arhar/Tur	2001	Kharif	100.0	Hectare	100.0	Tonnes	1.000000
...	...	...	...	...	...	...	...	...	...	...
4137	Punjab	RUPNAGAR	Wheat	2000	Rabi	86000.0	Hectare	312000.0	Tonnes	3.627907
4138	Punjab	SANGRUR	Wheat	1997	Rabi	401000.0	Hectare	1732000.0	Tonnes	4.319202
4139	Punjab	SANGRUR	Wheat	1998	Rabi	86000.0	Hectare	325000.0	Tonnes	3.779070
4140	Punjab	SANGRUR	Wheat	1999	Rabi	394000.0	Hectare	1902000.0	Tonnes	4.827411
4141	Punjab	SANGRUR	Wheat	2000	Rabi	393000.0	Hectare	1921000.0	Tonnes	4.888041

4142 rows × 10 columns

In [18]:

```
##Calculating the total production of various crops in Punjab since 1997, in tonnes.
punjab_crop_pro = df_pj.groupby(['Crop']).sum()[['Production']].reset_index()
punjab_crop_pro
```

Out[18]:

	Crop	Production
0	Arhar/Tur	118120.0
1	Bajra	78100.0
2	Barley	1393000.0
3	Cotton(lint)	34634200.0
4	Gram	97300.0
5	Groundnut	76500.0
6	Guar seed	369740.0
7	Jowar	400.0
8	Linseed	1700.0
9	Maize	10150400.0
10	Masoor	30979.0
11	Moong(Green Gram)	279490.0
12	Moth	1800.0
13	Other Rabi pulses	6300.0
14	Peas & beans (Pulses)	48620.0
15	Rapeseed &Mustard	1012100.0
16	Rice	243042000.0
17	Sesamum	71410.0
18	Sugarcane	125639000.0
19	Sunflower	96600.0

20	Urad	30350.0
21	Wheat	364370000.0
22	other oilseeds	3300.0

In [19]:

```
##Sorting to see which crops were produced more in Punjab since 1997
punjab_crop_pro.sort_values(by=['Production'])
```

Out[19]:

	Crop	Production
7	Jowar	400.0
8	Linseed	1700.0
12	Moth	1800.0
22	other oilseeds	3300.0
13	Other Rabi pulses	6300.0
20	Urad	30350.0
10	Masoor	30979.0
14	Peas & beans (Pulses)	48620.0
17	Sesamum	71410.0
5	Groundnut	76500.0
1	Bajra	78100.0
19	Sunflower	96600.0
4	Gram	97300.0
0	Arhar/Tur	118120.0
11	Moong(Green Gram)	279490.0
6	Guar seed	369740.0
15	Rapeseed &Mustard	1012100.0
2	Barley	1393000.0
9	Maize	10150400.0
3	Cotton(lint)	34634200.0
18	Sugarcane	125639000.0
16	Rice	243042000.0
21	Wheat	364370000.0

Wheat, rice, sugarcane, cotton and maize are the most cultivated crops in Punjab, with the least five being jowar, linseed, moth, other oilseeds and other rabi pulses.

In [20]:

```
##Creating a pie chart using matplotlib to vizualize the following crop distribution over th
```

In [21]:

```
##Exporting this data to a CSV.
punjab_crop_pro.to_csv('Crop production over years.csv')
```

In [22]:

```
punjab_crop_pro.astype({'Production': 'int'}).dtypes
```

Out[22]:

```
Crop           object
Production      int64
dtype: object
```

In [23]:

```
##Converting the columns to lists.
crops_name = punjab_crop_pro.Crop.to_list()
crops_name
```

```
crops_name
```

Out[23]:
```
['Arhar/Tur',
 'Bajra',
 'Barley',
 'Cotton(lint)',
 'Gram',
 'Groundnut',
 'Guar seed',
 'Jowar',
 'Linseed',
 'Maize',
 'Masoor',
 'Moong(Green Gram)',
 'Moth',
 'Other Rabi pulses',
 'Peas & beans (Pulses)',
 'Rapeseed &Mustard',
 'Rice',
 'Sesamum',
 'Sugarcane',
 'Sunflower',
 'Urad',
 'Wheat',
 'other oilseeds']
```

In [24]:
```
crops_production_total = punjab_crop_pro.Production.to_list()
crops_production_total
```

Out[24]:
```
[118120.0,
 78100.0,
 1393000.0,
 34634200.0,
 97300.0,
 76500.0,
 369740.0,
 400.0,
 1700.0,
 10150400.0,
 30979.0,
 279490.0,
 1800.0,
 6300.0,
 48620.0,
 1012100.0,
 243042000.0,
 71410.0,
 125639000.0,
 96600.0,
 30350.0,
 364370000.0,
 3300.0]
```

In [25]:
```
for element in crops_production_total:
    print (type(element))
```

```
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
```

```
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
```

In [26]:

```
##Using list comprehension to convert the list to int
crops_production_total = [int(x) for x in crops_production_total]
for element in crops_production_total:
    print (type(element))
```

```
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
```

In [27]:

```
##Converting this list to an array using numpy
crops_prod_array = np.array(crops_production_total)
crops_prod_array
```

Out[27]:

```
array([   118120,     78100,   1393000,  34634200,     97300,     76500,
          369740,       400,      1700,  10150400,     30979,    279490,
            1800,      6300,     48620,   1012100, 243042000,     71410,
       125639000,     96600,     30350, 364370000,      3300])
```

In [28]:

```
##Finding the sum of the array
sumproduction = np.sum(crops_prod_array)
print(sumproduction)
```

```
781551409
```

In [29]:

```
## Finding length of array
length = len(crops_prod_array)
print(length)
```

```
23
```

In [30]:

```
## Defining a function to calculate each crop percentage of total crop production over the y
percentlist = []
def percentile(x):
    for i in x, range (0, 22):
        croppercent = (i/sumproduction) * 100
        percentlist.extend(croppercent)

percentile(crops_production_total)
```

In [31]:
```
print(percentlist)
```

```
[0.015113529147255366, 0.00999294468676468, 0.17823523621847887, 4.431467924076124, 0.012449
596901692745, 0.009788223668854009, 0.047308468226432435, 5.1180254477668025e-05, 0.00021751
608153008909, 1.2987501376253037, 0.003963782758659194, 0.035760923309908585, 0.000230311145
14950607, 0.0008060890080232712, 0.006220959931760547, 0.12949883889211952, 31.0973785219034
8, 0.009136954930625683, 16.075589980799332, 0.012360031456356828, 0.003883301808493061, 46.
62137331006974, 0.0004222370994407612, 0.0, 1.2795063619417006e-07, 2.559012723883401e-07,
3.8385190858251014e-07, 5.118025447766802e-07, 6.397531809708502e-07, 7.677038171650203e-07,
8.956544533591903e-07, 1.0236050895533604e-06, 1.1515557257475304e-06, 1.2795063619417004e-0
6, 1.4074569981358706e-06, 1.5354076343300406e-06, 1.6633582705242108e-06, 1.791308906718380
5e-06, 1.919259542912551e-06, 2.047210179106721e-06, 2.175160815300891e-06, 2.30311145149506
1e-06, 2.4310620876892313e-06, 2.559012723883401e-06, 2.686963360077571e-06]
```

In [32]:
```
##Adding this percentage list to a new dataframe
##lists - percentlist, crops_name, crops_production_total

percentile_list_crops = pd.DataFrame(list(zip(crops_name,crops_production_total,percentlist)

percentile_list_crops.head(10)
```

Out[32]:

	Crop	Total Production	Percentage of total production
0	Arhar/Tur	118120	0.015114
1	Bajra	78100	0.009993
2	Barley	1393000	0.178235
3	Cotton(lint)	34634200	4.431468
4	Gram	97300	0.012450
5	Groundnut	76500	0.009788
6	Guar seed	369740	0.047308
7	Jowar	400	0.000051
8	Linseed	1700	0.000218
9	Maize	10150400	1.298750

In [33]:
```
##Sorting this
percentile_list_crops.sort_values(by=['Percentage of total production'])
```

Out[33]:

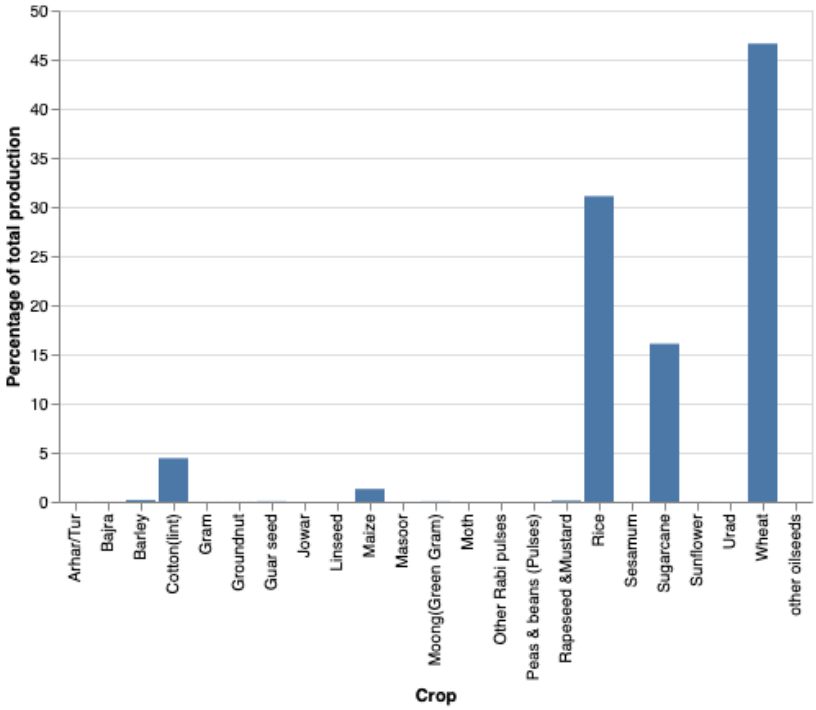
	Crop	Total Production	Percentage of total production
7	Jowar	400	0.000051
8	Linseed	1700	0.000218
12	Moth	1800	0.000230
22	other oilseeds	3300	0.000422
13	Other Rabi pulses	6300	0.000806
20	Urad	30350	0.003883
10	Masoor	30979	0.003964
14	Peas & beans (Pulses)	48620	0.006221
17	Sesamum	71410	0.009137
5	Groundnut	76500	0.009788
1	Bajra	78100	0.009993
19	Sunflower	96600	0.012360
4	Gram	97300	0.012450
0	Arhar/Tur	118120	0.015114
11	Moong(Green Gram)	279490	0.035761
6	Guar seed	369740	0.047308

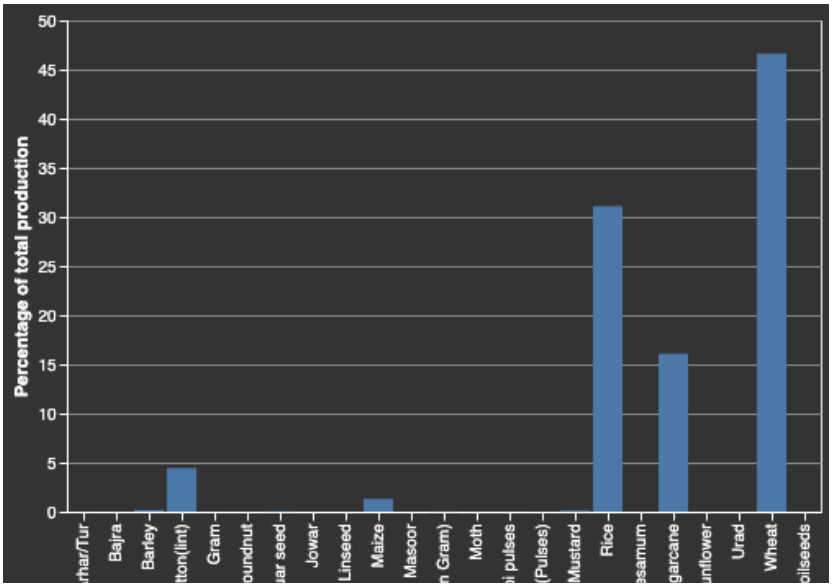
6	Guar seed	369740	0.047308
15	Rapeseed &Mustard	1012100	0.129499
2	Barley	1393000	0.178235
9	Maize	10150400	1.298750
3	Cotton(lint)	34634200	4.431468
18	Sugarcane	125639000	16.075590
16	Rice	243042000	31.097379
21	Wheat	364370000	46.621373

In [34]:

```
##Plotting a bar chart to represent the following information
percentilechart = alt.Chart(percentile_list_crops).mark_bar().encode(
    x='Crop',
    y='Percentage of total production'
)

percentilechart
```

Out[34]:





It is clear that compared to major crops such as wheat, rice, sugarcane and cotton, cultivation of other crops is largely miniscule. This can also be seen by the pie chart below:

In [35]:

```
# ##Using matplotlib to create a pie chart to show the same distribution of percentage
# ##plt.pie(percentlist)
# plt.axis('equal')
# plt.legend(crops_name, loc = 5)
# plt.show()
```

Wheat, rice, sugarcane and cotton are the major crops grown in Punjab.

In [36]:

```
df_pj_rice =  df[(df['State'] == 'Punjab') & (df['Crop'] == 'Rice')].reset_index(drop=True)
df_pj_rice
```

Out[36]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
0	Punjab	AMRITSAR	Rice	2001	Kharif	319000.0	Hectare	958000.0	Tonnes	3.003135
1	Punjab	AMRITSAR	Rice	2002	Kharif	311000.0	Hectare	879000.0	Tonnes	2.826367
2	Punjab	AMRITSAR	Rice	2003	Kharif	326000.0	Hectare	872000.0	Tonnes	2.674847
3	Punjab	BATHINDA	Rice	2001	Kharif	82000.0	Hectare	307000.0	Tonnes	3.743902
4	Punjab	BATHINDA	Rice	2002	Kharif	107000.0	Hectare	367000.0	Tonnes	3.429907
...	...	...	...	...	...	...	...	...	...	...
443	Punjab	RUPNAGAR	Rice	2000	Kharif	49000.0	Hectare	163000.0	Tonnes	3.326531
444	Punjab	SANGRUR	Rice	1997	Kharif	333000.0	Hectare	1277000.0	Tonnes	3.834835
445	Punjab	SANGRUR	Rice	1998	Kharif	353000.0	Hectare	1262000.0	Tonnes	3.575071
446	Punjab	SANGRUR	Rice	1999	Kharif	360000.0	Hectare	1282000.0	Tonnes	3.561111
447	Punjab	SANGRUR	Rice	2000	Kharif	357000.0	Hectare	1342000.0	Tonnes	3.759104

448 rows × 10 columns

In [37]:

```
pj_rice_prod = df_pj_rice.groupby(['Year']).sum()[['Production']].reset_index()
pj_rice_prod
```

Out[37]:

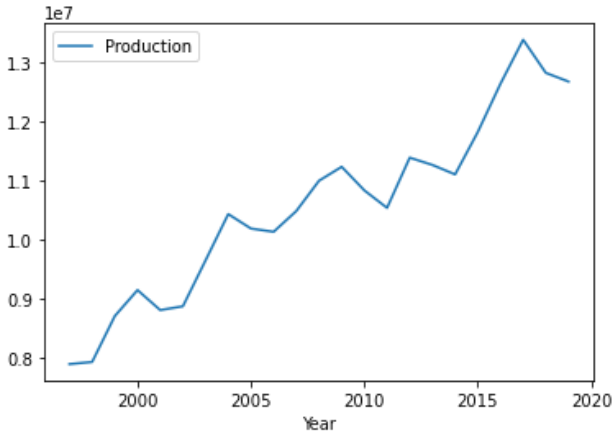
	Year	Production
0	1997	7904000.0
1	1998	7940000.0
2	1999	8716000.0
3	2000	9154000.0
4	2001	8816000.0
5	2002	8880000.0
6	2003	9656000.0
7	2004	10437000.0
8	2005	10193000.0
9	2006	10138000.0
10	2007	10489000.0
11	2008	11000000.0

12	2009	11236000.0
13	2010	10837000.0
14	2011	10542000.0
15	2012	11390000.0
16	2013	11267000.0
17	2014	11107000.0
18	2015	11823000.0
19	2016	12638000.0
20	2017	13382000.0
21	2018	12822000.0
22	2019	12675000.0

In [38]:

```
pj_rice_prod.plot.line(x='Year', y='Production')
```

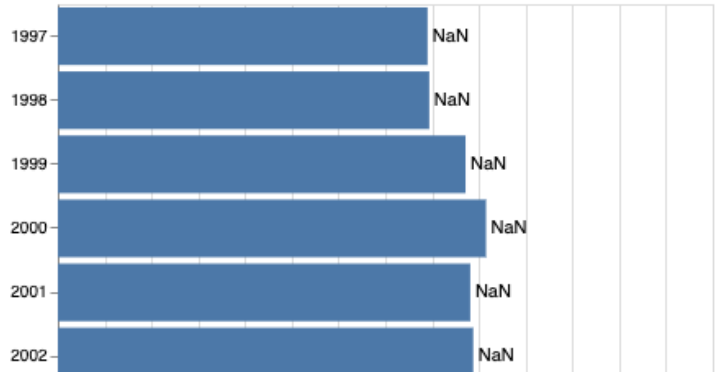
Out[38]: <AxesSubplot:xlabel='Year'>

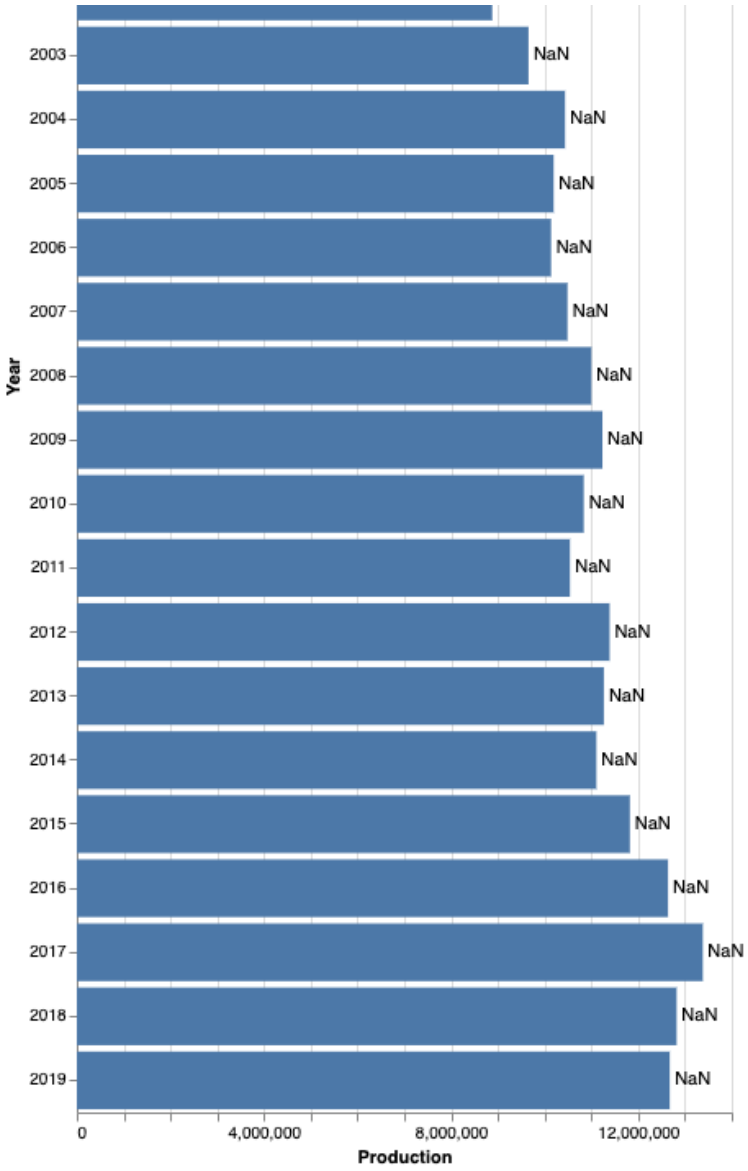


In [39]:

```
source =  pj_rice_prod

bars = alt.Chart(source).mark_bar().encode(
    x='Production:Q',
    y="Year:O"
)

text = bars.mark_text(
    align='left',
    baseline='middle',
    dx=3  # Nudges text to right so it doesn't appear on top of the bar
).encode(
    text='wheat:Q'
)

(bars + text).properties(height=900)
```

Out[39]:



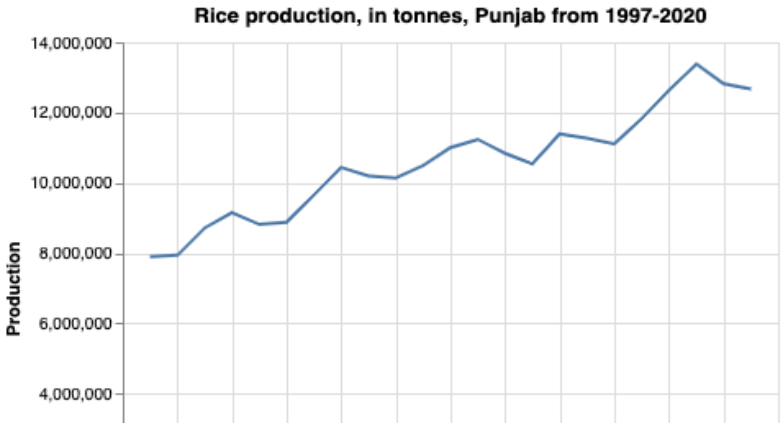


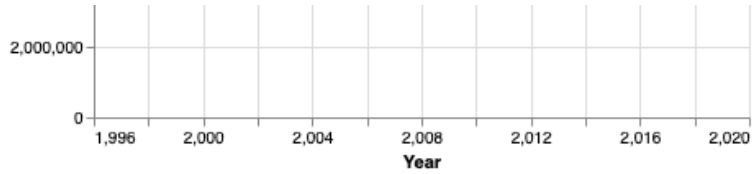
For all the data viz developed using Altair, I will be dropping in the image files into the markdown so that it can render on Github.

In [40]:

```
alt.Chart(pj_rice_prod).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Punjab from 1997-2020')
```

Out[40]:





In [41]:

```
df_pj_rice  = df.loc[(df['State'] == 'Punjab') & (df['Crop'] == 'Rice')] ## I am subsetting
df_pj_rice.head(20)
```

Out[41]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
29303	Punjab	AMRITSAR	Rice	2001	Kharif	319000.0	Hectare	958000.0	Tonnes	3.003135
29304	Punjab	AMRITSAR	Rice	2002	Kharif	311000.0	Hectare	879000.0	Tonnes	2.826367
29305	Punjab	AMRITSAR	Rice	2003	Kharif	326000.0	Hectare	872000.0	Tonnes	2.674847
29306	Punjab	BATHINDA	Rice	2001	Kharif	82000.0	Hectare	307000.0	Tonnes	3.743902
29307	Punjab	BATHINDA	Rice	2002	Kharif	107000.0	Hectare	367000.0	Tonnes	3.429907
29308	Punjab	BATHINDA	Rice	2003	Kharif	105000.0	Hectare	419000.0	Tonnes	3.990476
29309	Punjab	FARIDKOT	Rice	2001	Kharif	70000.0	Hectare	267000.0	Tonnes	3.814286
29310	Punjab	FARIDKOT	Rice	2002	Kharif	86000.0	Hectare	280000.0	Tonnes	3.255814
29311	Punjab	FARIDKOT	Rice	2003	Kharif	84000.0	Hectare	308000.0	Tonnes	3.666667
29312	Punjab	FATEHGARH SAHIB	Rice	2001	Kharif	81000.0	Hectare	324000.0	Tonnes	4.000000
29313	Punjab	FATEHGARH SAHIB	Rice	2002	Kharif	80000.0	Hectare	312000.0	Tonnes	3.900000
29314	Punjab	FATEHGARH SAHIB	Rice	2003	Kharif	83000.0	Hectare	364000.0	Tonnes	4.385542
29315	Punjab	FIROZEPUR	Rice	2001	Kharif	230000.0	Hectare	864000.0	Tonnes	3.756522
29316	Punjab	FIROZEPUR	Rice	2002	Kharif	234000.0	Hectare	824000.0	Tonnes	3.521368
29317	Punjab	FIROZEPUR	Rice	2003	Kharif	244000.0	Hectare	903000.0	Tonnes	3.700820
29318	Punjab	GURDASPUR	Rice	2001	Kharif	194000.0	Hectare	571000.0	Tonnes	2.943299
29319	Punjab	GURDASPUR	Rice	2002	Kharif	189000.0	Hectare	547000.0	Tonnes	2.894180
29320	Punjab	GURDASPUR	Rice	2003	Kharif	193000.0	Hectare	587000.0	Tonnes	3.041451
29321	Punjab	HOSHIARPUR	Rice	2001	Kharif	60000.0	Hectare	171000.0	Tonnes	2.850000
29322	Punjab	HOSHIARPUR	Rice	2002	Kharif	56000.0	Hectare	159000.0	Tonnes	2.839286

In [42]:

```
df_pj_rice.District.unique() ## To find the various districts where rice is grown in Punjab.
```

Out[42]:

```
array(['AMRITSAR', 'BATHINDA', 'FARIDKOT', 'FATEHGARH SAHIB', 'FIROZEPUR',
       'GURDASPUR', 'HOSHIARPUR', 'JALANDHAR', 'KAPURTHALA', 'LUDHIANA',
       'MANSA', 'MOGA', 'MUKTSAR', 'NAWANSHAHR', 'PATIALA', 'RUPNAGAR',
       'SANGRUR', 'S', 'TARN TARAN', 'BARNALA', 'FAZILKA', 'PATHANKOT'],
      dtype=object)
```

Across districts of Punjab

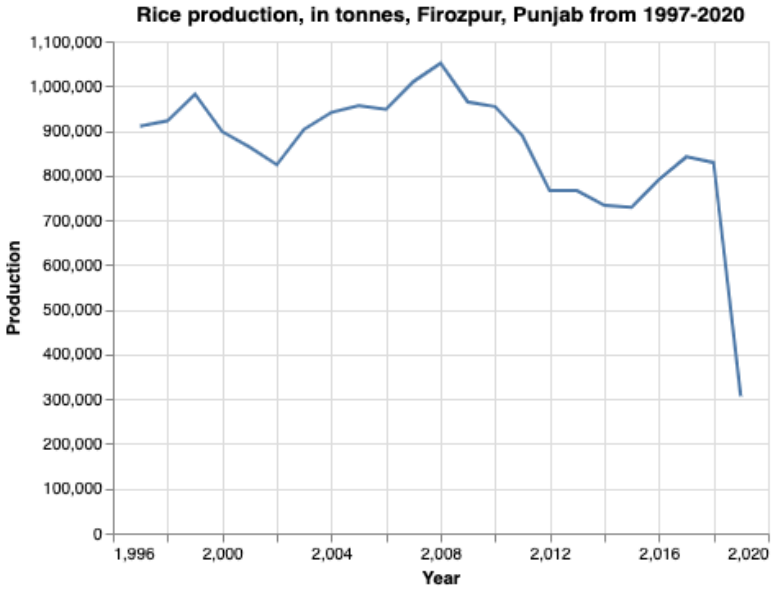
Punjab has 22 districts, of which 21 are represented here clearly.

Firozpur

In [43]:

```
df_pj_fi = df_pj_rice.loc[df['District'] == 'FIROZEPUR']

alt.Chart(df_pj_fi).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Firozpur, Punjab from 1997-2020')
```

Out[43]:



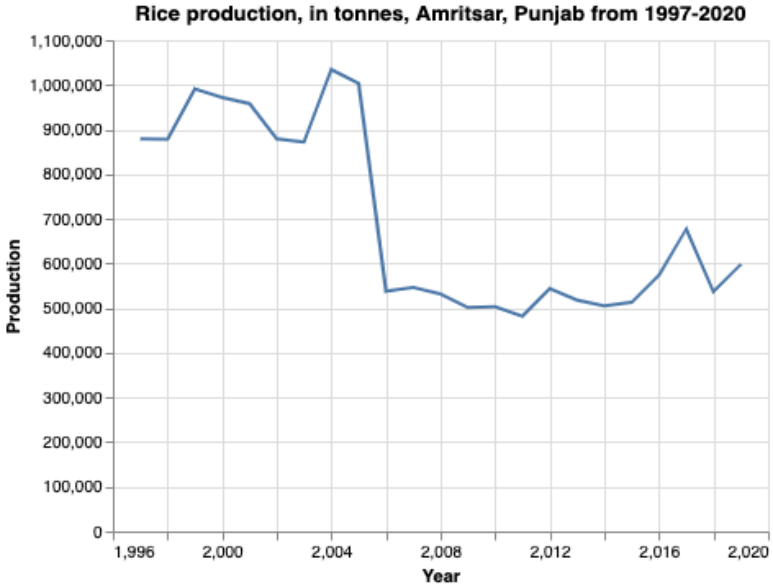
There really has been a considerable decline of rice production in this district. This may be true for other districts also.

Amritsar

In [44]:

```
df_pj_am = df_pj_rice.loc[df['District'] == 'AMRITSAR']

alt.Chart(df_pj_am).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Amritsar, Punjab from 1997-2020')
```

Out[44]:



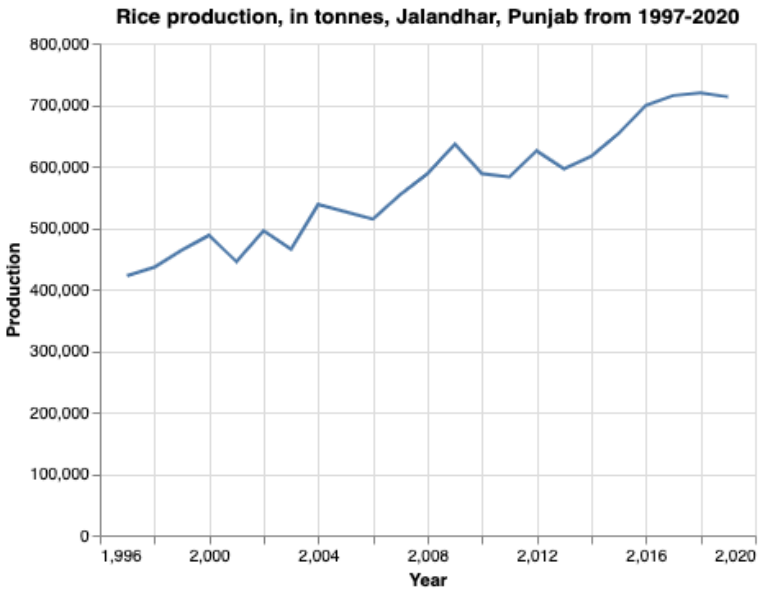
JALANDHAR

A marked decline here as well.

In [45]:

```
df_pj_ja = df_pj_rice.loc[df['District'] == 'JALANDHAR']

alt.Chart(df_pj_ja).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Jalandhar, Punjab from 1997-2020')
```

Out[45]:

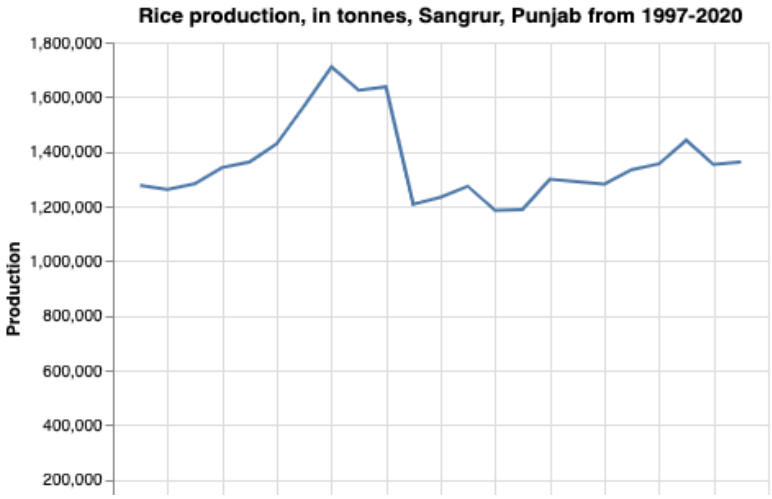


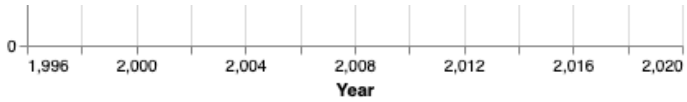
SANGRUR

In [46]:

```
df_pj_sa = df_pj_rice.loc[df['District'] == 'SANGRUR']

alt.Chart(df_pj_sa).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Sangrur, Punjab from 1997-2020')
```

Out[46]:



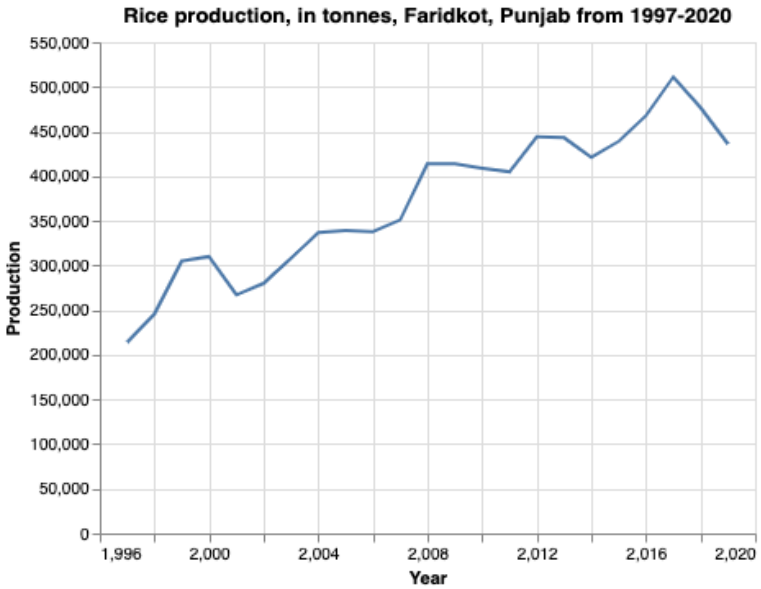


FARIDKOT

In [47]:

```
df_pj_far = df_pj_rice.loc[df['District'] == 'FARIDKOT']

alt.Chart(df_pj_far).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, Faridkot, Punjab from 1997-2020')
```

Out[47]:

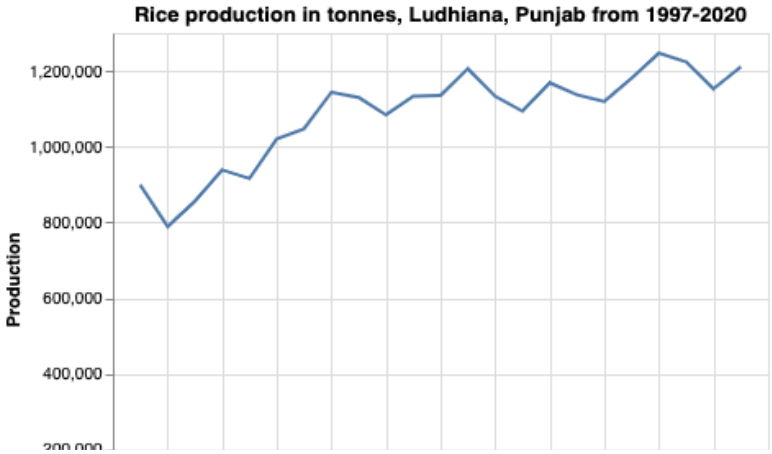


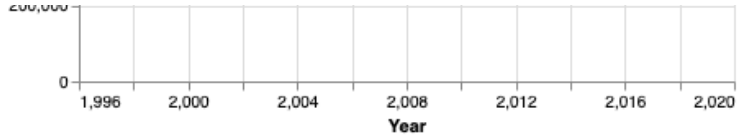
LUDHIANA

In [48]:

```
df_pj_lud = df_pj_rice.loc[df['District'] == 'LUDHIANA']

alt.Chart(df_pj_lud).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production in tonnes, Ludhiana, Punjab from 1997-2020')
```

Out[48]:





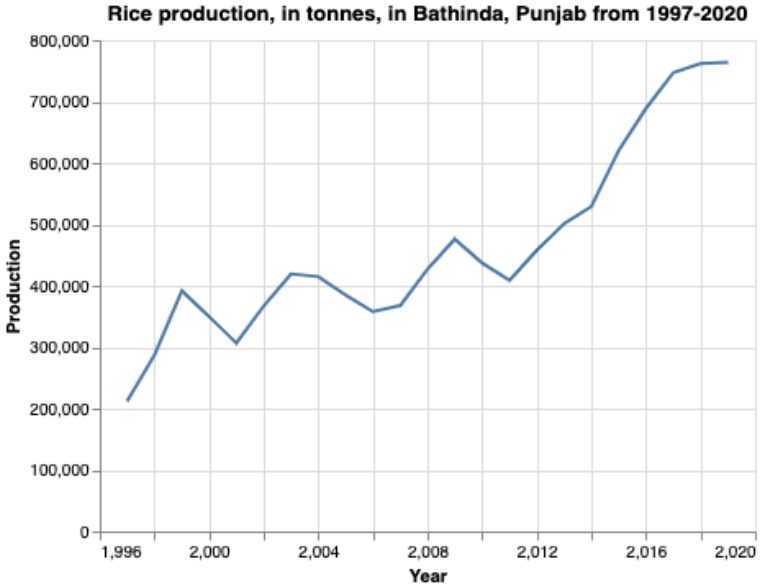
The district of Ludhiana does not fall in this trend as dramatically as the other districts. Yet, the production has fluctuated over time.

BATHINDA

In [49]:

```
df_pj_bat = df_pj_rice.loc[df['District'] == 'BATHINDA']

alt.Chart(df_pj_bat).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Bathinda, Punjab from 1997-2020')
```

Out[49]:



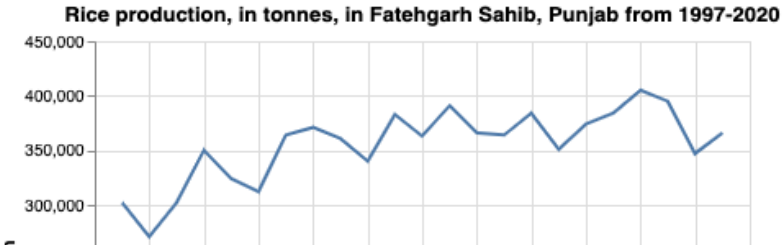
Bathinda seems similar to Ludhiana.

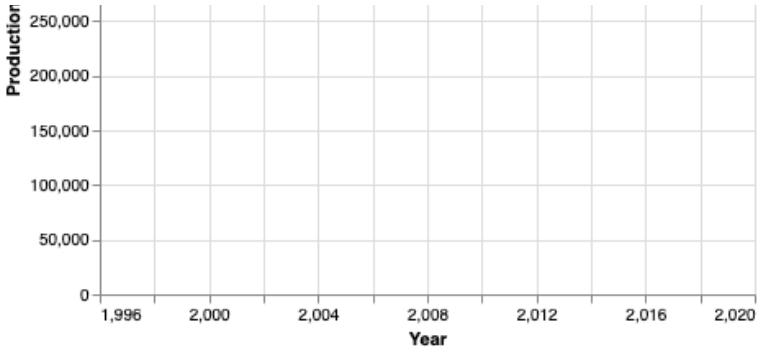
FATEHGARH SAHIB

In [50]:

```
df_pj_fs = df_pj_rice.loc[df['District'] == 'FATEHGARH SAHIB']

alt.Chart(df_pj_fs).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Fatehgarh Sahib, Punjab from 1997-2020')
```

Out[50]:



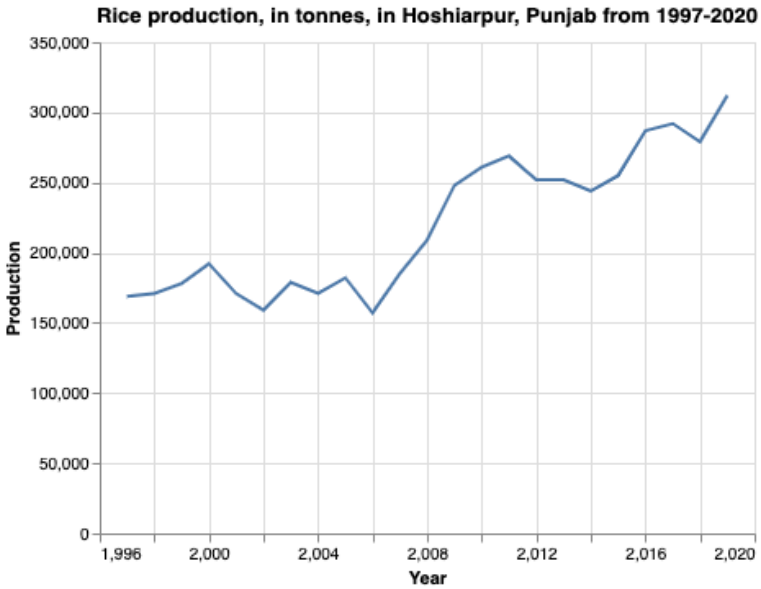


HOSHIARPUR

In [51]:

```
df_pj_hos = df_pj_rice.loc[df['District'] == 'HOSHIARPUR']

alt.Chart(df_pj_hos).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Hoshiarpur, Punjab from 1997-2020')
```

Out[51]:

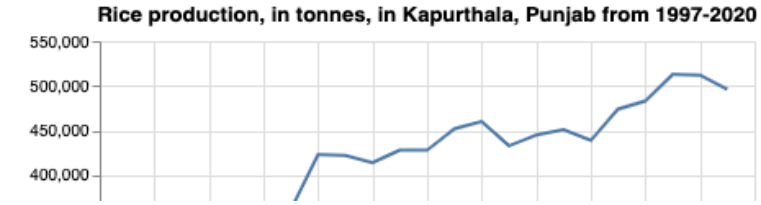


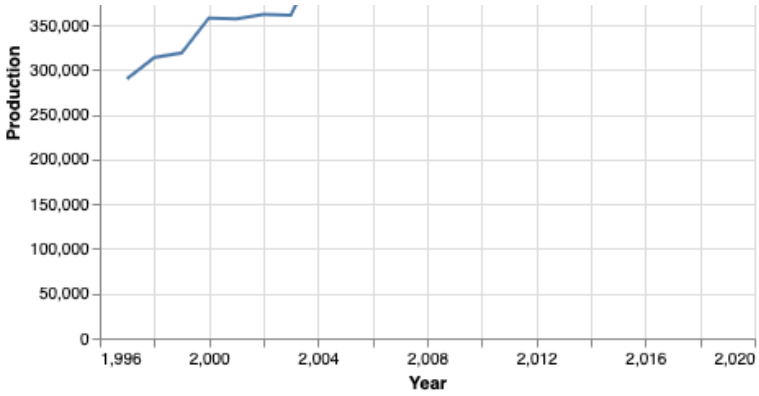
KAPURTHALA

In [52]:

```
df_pj_kap = df_pj_rice.loc[df['District'] == 'KAPURTHALA']

alt.Chart(df_pj_kap).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Kapurthala, Punjab from 1997-2020')
```

Out[52]:



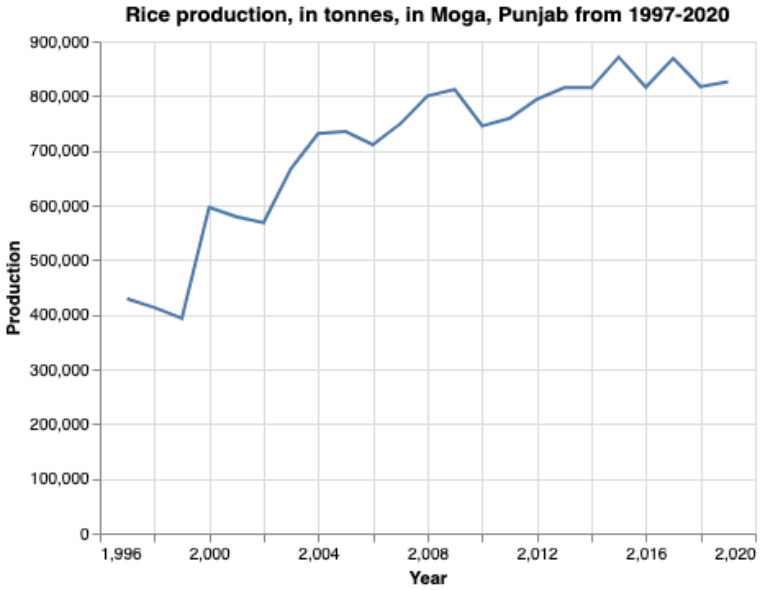


MOGA

In [53]:

```
df_pj_mog = df_pj_rice.loc[df['District'] == 'MOGA']

alt.Chart(df_pj_mog).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Moga, Punjab from 1997-2020')
```

Out[53]:

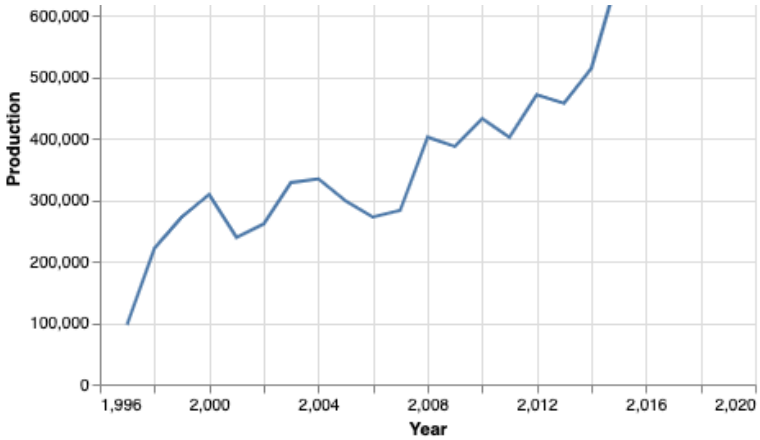


MUKTSAR

In [54]:

```
df_pj_muk = df_pj_rice.loc[df['District'] == 'MUKTSAR']

alt.Chart(df_pj_muk).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Muktsar, Punjab from 1997-2020')
```

Out[54]:

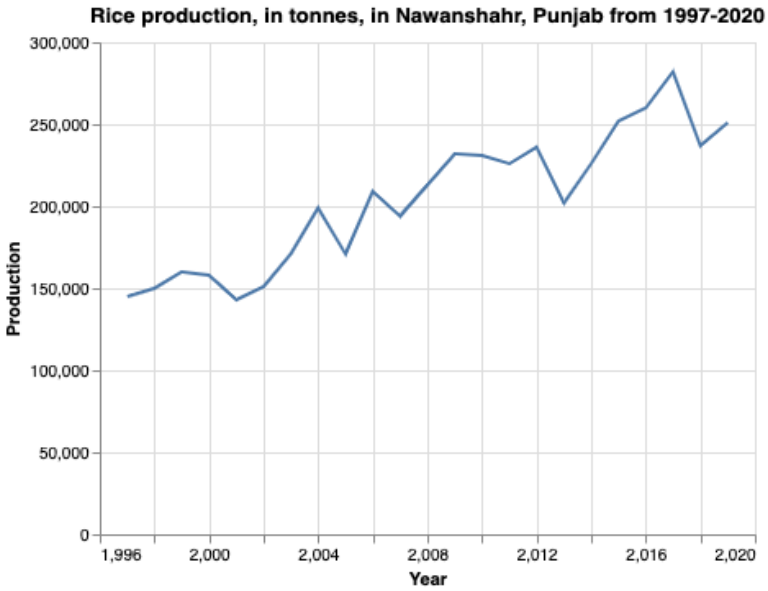


NAWANSHAHR

In [55]:

```
df_pj_naw = df_pj_rice.loc[df['District'] == 'NAWANSHAHR']

alt.Chart(df_pj_naw).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Nawanshahr, Punjab from 1997-2020')
```

Out[55]:

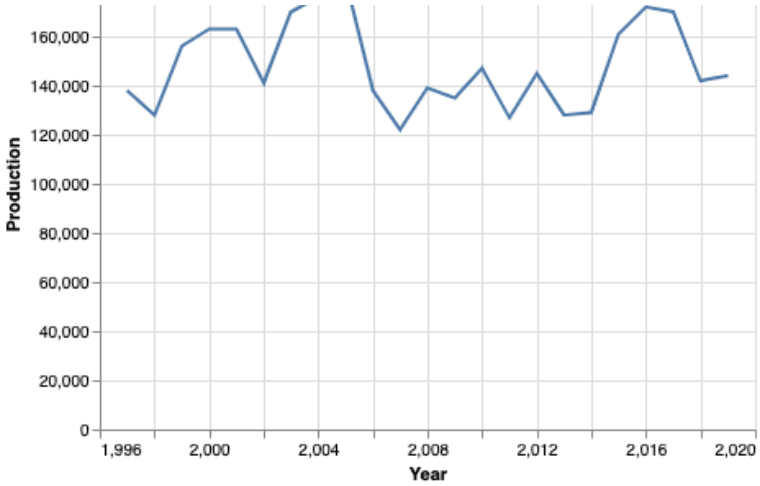


RUPNAGAR

In [56]:

```
df_pj_rup = df_pj_rice.loc[df['District'] == 'RUPNAGAR']

alt.Chart(df_pj_rup).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Rupnagar, Punjab from 1997-2020')
```

Out[56]:

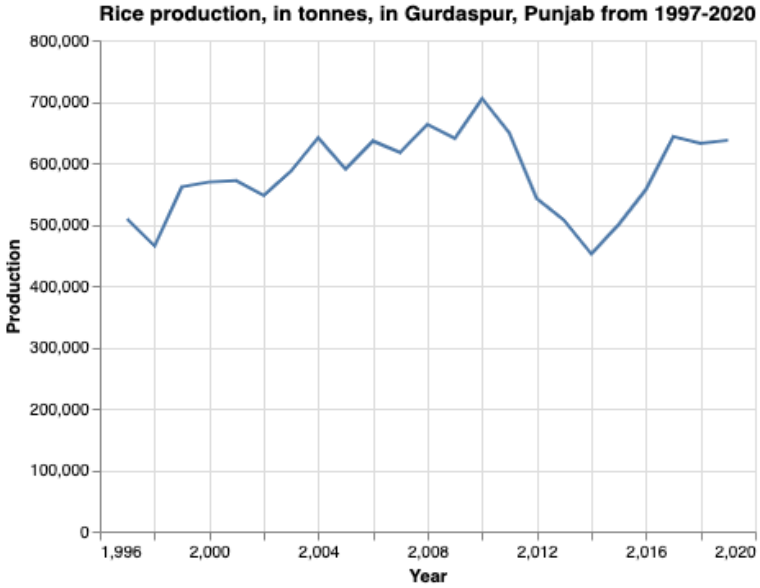


GURDASPUR

In [57]:

```
df_pj_gur = df_pj_rice.loc[df['District'] == 'GURDASPUR']

alt.Chart(df_pj_gur).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Gurdaspur, Punjab from 1997-2020')
```

Out[57]:

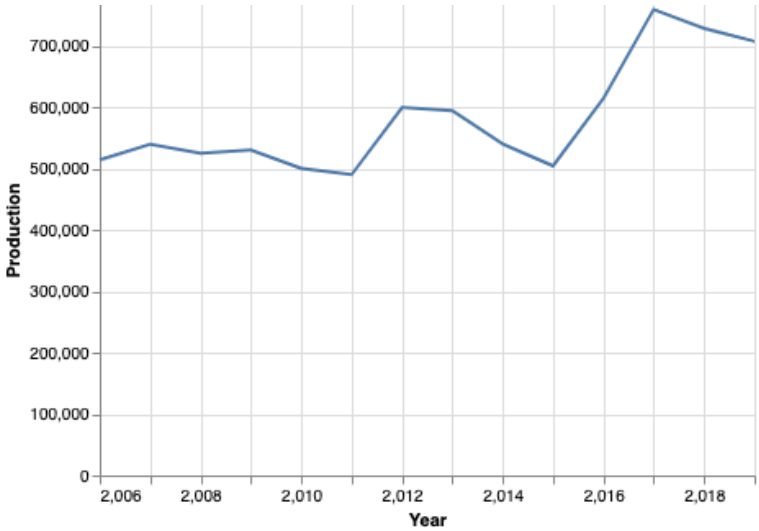


TARN TARAN

In [58]:

```
df_pj_tt = df_pj_rice.loc[df['District'] == 'TARN TARAN']

alt.Chart(df_pj_tt).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Tarn Taran, Punjab from 1997-2020')
```

Out[58]:

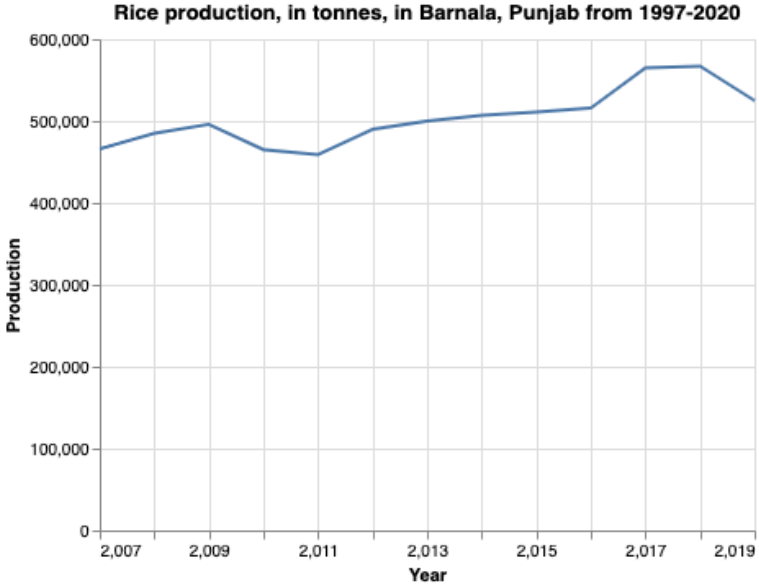


BARNALA

In [59]:

```
df_pj_bar = df_pj_rice.loc[df['District'] == 'BARNALA']

alt.Chart(df_pj_bar).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Barnala, Punjab from 1997-2020')
```

Out[59]:



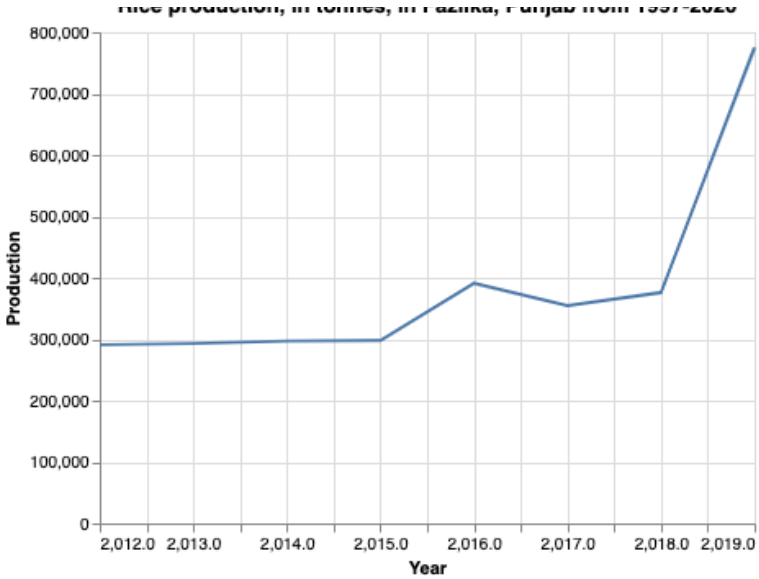
FAZILKA

In [60]:

```
df_pj_faz = df_pj_rice.loc[df['District'] == 'FAZILKA']

alt.Chart(df_pj_faz).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Fazilka, Punjab from 1997-2020')
```

Rice production, in tonnes, in Fazilka, Punjab from 1997-2020

Out[60]:

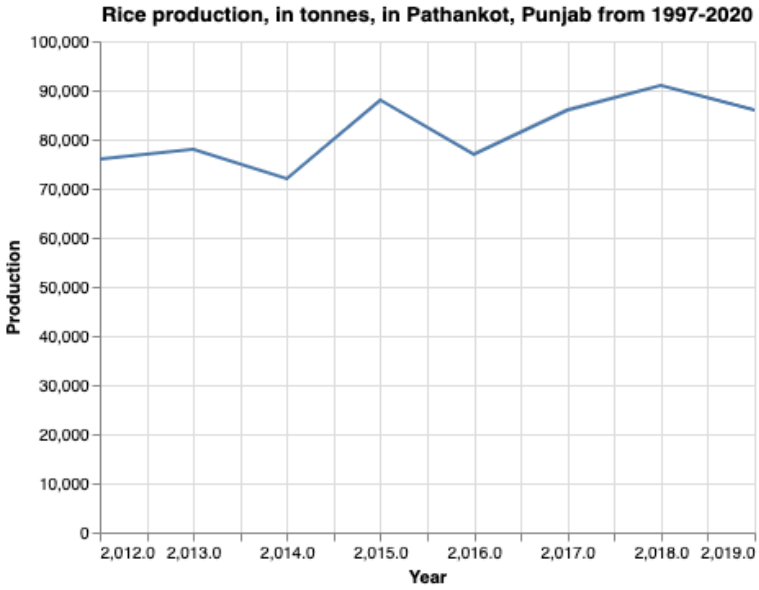


PATHANKOT

In [61]:

```
df_pj_pat = df_pj_rice.loc[df['District'] == 'PATHANKOT']

alt.Chart(df_pj_pat).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in Pathankot, Punjab from 1997-2020')
```

Out[61]:



S

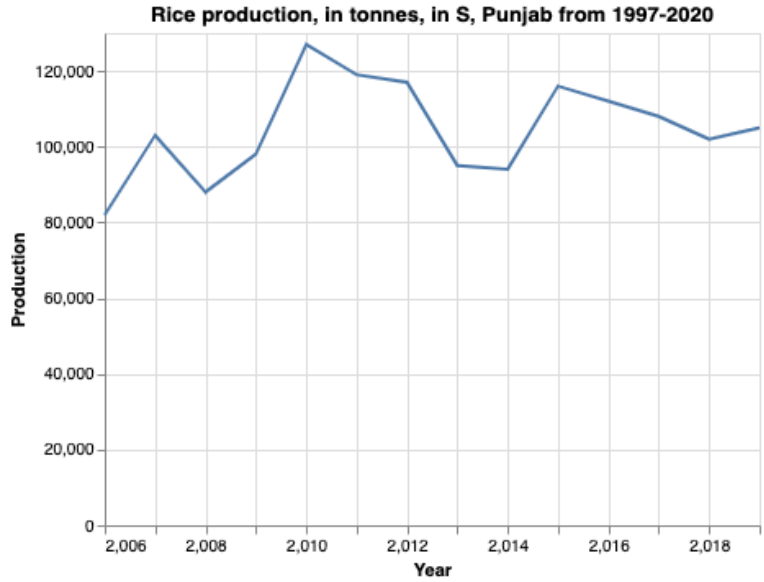
It is not known which district this represents. It is only stated as 'S'. Here are a list of districts in Punjab. It seems like this 'S' can be either Sahibzada Ajit Singh Nagar, or Mansa. Since it begins with an 'S', I am guessing it is the former, but for now, let us leave it at S. Except for one district, every other district of Punjab has been represented by this data.

In [62]:

```
df_pj_s = df_pj_rice.loc[df['District'] == 'S']
```

```
alt.Chart(df_pj_s).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rice production, in tonnes, in S, Punjab from 1997-2020')
```

Out[62]:



Except for the districts of Fazilka, Hoshiarpur and Bathinda, the other districts of Punjab are showing moderate to steep decline in rice production over the past 5-10 years, especially. This seems to correspond with the earlier hypothesis of the diversification of the crops in Punjab.

Now, I wish to compare the trajectories of the various crops grown in Punjab with the other crops. For this, I would need to use groupby to select the various parameters and then use Altair to create line charts.

Comparing the production of rice in Punjab over time with other crops

In [63]:

```
# subset for Punjab and Rice
rice = production_all[
    (production_all['State'] == 'Punjab') &
    (production_all['Crop'] == 'Rice')
].reset_index()
```

In [64]:

```
rice.head(10)
```

Out[64]:

	index	State	Crop	Year	Season	Production
0	15849	Punjab	Rice	1997	Kharif	7904000.0
1	15850	Punjab	Rice	1998	Kharif	7940000.0
2	15851	Punjab	Rice	1999	Kharif	8716000.0
3	15852	Punjab	Rice	2000	Kharif	9154000.0
4	15853	Punjab	Rice	2001	Kharif	8816000.0
5	15854	Punjab	Rice	2002	Kharif	8880000.0
6	15855	Punjab	Rice	2003	Kharif	9656000.0
7	15856	Punjab	Rice	2004	Kharif	10437000.0
8	15857	Punjab	Rice	2005	Kharif	10193000.0

9	15858	Punjab	Rice	2006	Kharif	10138000.0

In [65]:

```
rice.to_csv('Ricepunjab.csv')
```

In [66]:

```
##Subset for Punjab and Wheat
wheat = production_all[
    (production_all['State'] == 'Punjab') &
    (production_all['Crop'] == 'Wheat')
].reset_index()

wheat.to_csv('wheatpj.csv')
```

In [67]:

```
wheat.head(50)
```

Out[67]:

	index	State	Crop	Year	Season	Production
0	15941	Punjab	Wheat	1997	Rabi	12715000.0
1	15942	Punjab	Wheat	1998	Rabi	14460000.0
2	15943	Punjab	Wheat	1999	Rabi	15910000.0
3	15944	Punjab	Wheat	2000	Rabi	15551000.0
4	15945	Punjab	Wheat	2001	Rabi	15499000.0
5	15946	Punjab	Wheat	2002	Rabi	14175000.0
6	15947	Punjab	Wheat	2003	Rabi	14489000.0
7	15948	Punjab	Wheat	2004	Rabi	14698000.0
8	15949	Punjab	Wheat	2005	Rabi	14493000.0
9	15950	Punjab	Wheat	2006	Rabi	14596000.0
10	15951	Punjab	Wheat	2007	Rabi	15720000.0
11	15952	Punjab	Wheat	2008	Rabi	15733000.0
12	15953	Punjab	Wheat	2009	Rabi	15169000.0
13	15954	Punjab	Wheat	2010	Rabi	16472000.0
14	15955	Punjab	Wheat	2011	Rabi	17982000.0
15	15956	Punjab	Wheat	2012	Rabi	16614000.0
16	15957	Punjab	Wheat	2013	Rabi	17620000.0
17	15958	Punjab	Wheat	2014	Rabi	15050000.0
18	15959	Punjab	Wheat	2015	Rabi	16077000.0
19	15960	Punjab	Wheat	2016	Rabi	17636000.0
20	15961	Punjab	Wheat	2017	Rabi	17830000.0
21	15962	Punjab	Wheat	2018	Rabi	18262000.0
22	15963	Punjab	Wheat	2019	Rabi	17619000.0

In [68]:

```
##Subset for Punjab and Wheat
bajra = production_all[
    (production_all['State'] == 'Punjab') &
    (production_all['Crop'] == 'Bajra')
].reset_index()
bajra.to_csv('bajrapj.csv')
```

In [69]:

```
bajra.head(50)
```

Out[69]:

	index	State	Crop	Year	Season	Production

0	15591	Punjab	Bajra	1997	Kharif	8000.0
1	15592	Punjab	Bajra	1998	Kharif	4000.0
2	15593	Punjab	Bajra	1999	Kharif	4000.0
3	15594	Punjab	Bajra	2000	Kharif	5000.0
4	15595	Punjab	Bajra	2002	Kharif	6000.0
5	15596	Punjab	Bajra	2003	Kharif	8000.0
6	15597	Punjab	Bajra	2004	Kharif	7000.0
7	15598	Punjab	Bajra	2005	Kharif	5000.0
8	15599	Punjab	Bajra	2006	Kharif	6000.0
9	15600	Punjab	Bajra	2007	Kharif	4000.0
10	15601	Punjab	Bajra	2008	Kharif	5000.0
11	15602	Punjab	Bajra	2009	Kharif	4000.0
12	15603	Punjab	Bajra	2010	Kharif	3000.0
13	15604	Punjab	Bajra	2011	Kharif	3000.0
14	15605	Punjab	Bajra	2012	Kharif	3000.0
15	15606	Punjab	Bajra	2013	Kharif	800.0
16	15607	Punjab	Bajra	2016	Kharif	700.0
17	15608	Punjab	Bajra	2017	Kharif	600.0
18	15609	Punjab	Bajra	2018	Kharif	700.0
19	15610	Punjab	Bajra	2019	Kharif	300.0

In [70]:

```
wheat_production_list = list(wheat["Production"])
print(wheat_production_list)
```

```
[12715000.0, 14460000.0, 15910000.0, 15551000.0, 15499000.0, 14175000.0, 14489000.0, 1469800
0.0, 14493000.0, 14596000.0, 15720000.0, 15733000.0, 15169000.0, 16472000.0, 17982000.0, 166
14000.0, 17620000.0, 15050000.0, 16077000.0, 17636000.0, 17830000.0, 18262000.0, 17619000.0]
```

In [71]:

```
rice_production_list = list(rice["Production"])
print(rice_production_list)
```

```
[7904000.0, 7940000.0, 8716000.0, 9154000.0, 8816000.0, 8880000.0, 9656000.0, 10437000.0, 10
193000.0, 10138000.0, 10489000.0, 11000000.0, 11236000.0, 10837000.0, 10542000.0, 11390000.
0, 11267000.0, 11107000.0, 11823000.0, 12638000.0, 13382000.0, 12822000.0, 12675000.0]
```

In [72]:

```
bajra_production_list = list(bajra["Production"])
print(bajra_production_list)
```

```
[8000.0, 4000.0, 4000.0, 5000.0, 6000.0, 8000.0, 7000.0, 5000.0, 6000.0, 4000.0, 5000.0, 400
0.0, 3000.0, 3000.0, 3000.0, 800.0, 700.0, 600.0, 700.0, 300.0]
```

I have already checked and seen that for all three crop csvs, the years column starts with 1997-98.

In [73]:

```
years_list = list(wheat["Year"])
print(years_list)
```

```
[1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2
012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]
```

Finding mean values of all the three crops' production over the years

To find the mean value, I will use numpy, and convert the following lists above to numpy arrays.

In [74]:

```
wheat_array = np.array(wheat_production_list, dtype = int)
```

```
wheat_array = np.array(wheat_production_list, dtype = int)
print(wheat_array)
```

```
[12715000 14460000 15910000 15551000 15499000 14175000 14489000 14698000
 14493000 14596000 15720000 15733000 15169000 16472000 17982000 16614000
 17620000 15050000 16077000 17636000 17830000 18262000 17619000]
```

In [75]:

```
rice_array = np.array(rice_production_list, dtype = int)
print(rice_array)
```

```
[ 7904000  7940000  8716000  9154000  8816000  8880000  9656000 10437000
 10193000 10138000 10489000 11000000 11236000 10837000 10542000 11390000
 11267000 11107000 11823000 12638000 13382000 12822000 12675000]
```

In [76]:

```
bajra_array = np.array(bajra_production_list, dtype = int)
print(bajra_array)
```

```
[8000 4000 4000 5000 6000 8000 7000 5000 6000 4000 5000 4000 3000 3000
 3000  800  700  600  700  300]
```

Using np.mean(), I will explore this data.

Mean

In [77]:

```
## Wheat
wheat_mean = np.mean(wheat_array)
print(wheat_mean)
```

```
15842173.913043479
```

In [78]:

```
#Rice
rice_mean = np.mean(rice_array)
print(rice_mean)
```

```
10567043.47826087
```

In [79]:

```
#Bajra
bajra_mean = np.mean(bajra_array)
print(bajra_mean)
```

```
3905.0
```

Even though Punjab it is a leadeing rice-growing state, it is evident from the mean values that the production of wheat is higher. To check this, lets use np.sum() just to verify.

In [80]:

```
rice_sum = np.sum(rice_array)
wheat_sum = np.sum(wheat_array)
print(rice_sum)
print(wheat_sum)
```

```
243042000
364370000
```

In [81]:

```
difference = wheat_sum - rice_sum
print(difference)
```

```
121328000
```

It is clear that the tonnes of wheat produced in the state since 1997, with 121328000 more tonnes of wheat produced over rice over this time period.

Minimum and maximum values

In [82]:

```
##Minimum and maxmimum amount of rice produced since 1997
```

In [83]:
```
rice_min = np.amin(rice_array)
print(rice_min)
```

```
7904000
```

In [84]:
```
rice_max = np.amax(rice_array)
print(rice_max)
```

```
13382000
```

In [85]:
```
difference_rice = rice_max - rice_min
print(difference_rice)
```

```
5478000
```

Rice production in Punjab fluctuates, as the difference between max and min production values is a large 5478000.

In [86]:
```
(difference_rice/rice_mean)*100
```

Out[86]:
```
51.840422642999975
```

This value is more than half of the value of the mean value of production.

For wheat, the same calculations:

In [87]:
```
wheat_min = np.amin(wheat_array)
print(wheat_min)
wheat_max = np.amax(wheat_array)
print(wheat_max)
difference_wheat = wheat_max - wheat_min
print(difference_wheat)
(difference_wheat/wheat_mean)*100
```

```
12715000
18262000
5547000
```

Out[87]:
```
35.0141339846859
```

It seems like wheat follows a similar pattern, but its production seems to fluctuate less than that of rice.

For bajra, the same calculations:

In [88]:
```
bajra_min = np.amin(bajra_array)
print(bajra_min)
bajra_max = np.amax(bajra_array)
print(bajra_max)
difference_bajra = bajra_max - bajra_min
print(difference_bajra)
(difference_bajra/bajra_mean)*100
```

```
300
8000
7700
```

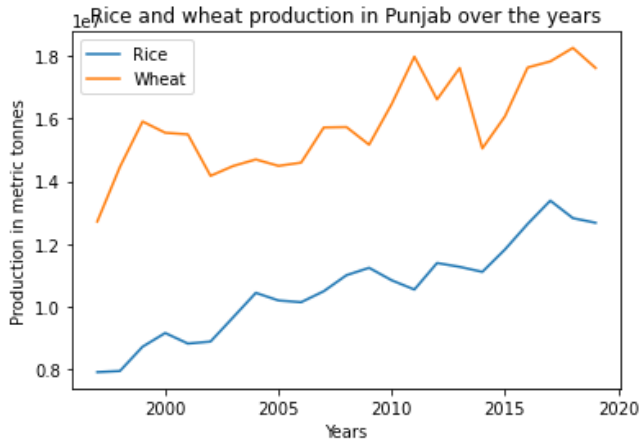
Out[88]:
```
197.1830985915493
```

Here, it seems like the production of bajra in the state has fluctuated immensely over the years. Lets plot the production of the three crops over the same time period in one line graph, using matplotlib.

Crop production of rice and wheat over the years

In [89]:
```
from matplotlib import pyplot as plt
```

In [90]:
```
plt.plot(years_list, rice_array)
plt.plot(years_list, wheat_array)
plt.legend(["Rice", "Wheat"])
plt.xlabel('Years')
plt.ylabel('Production in metric tonnes')
plt.title('Rice and wheat production in Punjab over the years')
plt.show()
```



The production of wheat and rice in Punjab, which seems to peak between 2015-2020, has been declining over the past few years.

The rise in the past 10 years or so in the production of rice can be linked to the higher levels of stubble burning also, though not directly so. This is because as per this article higher levels of farm mechanizations and increase in landholdings, especially under paddy, has led to the rise of stubble burning. This has had serious consequences on the air pollution levels east of Punjab, an effect clearly seen in Delhi NCR. Even so, this year, Punjab recorded the highest levels of stubble burning, as per this article, and the factors of increasing pollution can be said to be indirectly linked to higher levels of production of paddy crops.

Exploring the trends of other crops in Punjab

In [91]:
```
production_all.head(10)
```

Out[91]:

	State	Crop	Year	Season	Production
0	Andaman and Nicobar Islands	Arecanut	2000	Kharif	7200.00
1	Andaman and Nicobar Islands	Arecanut	2001	Kharif	7300.00
2	Andaman and Nicobar Islands	Arecanut	2002	Whole Year	7350.00
3	Andaman and Nicobar Islands	Arecanut	2003	Whole Year	6707.00
4	Andaman and Nicobar Islands	Arecanut	2004	Whole Year	4781.05
5	Andaman and Nicobar Islands	Arecanut	2005	Whole Year	3058.46
6	Andaman and Nicobar Islands	Arecanut	2006	Whole Year	5839.30
7	Andaman and Nicobar Islands	Arecanut	2007	Kharif	3415.44
8	Andaman and Nicobar Islands	Arecanut	2007	Rabi	2276.96
9	Andaman and Nicobar Islands	Arecanut	2008	Autumn	3060.00

In [92]:
```
pj  = df.loc[(df['State'] == 'Punjab')].reset_index(drop=True)
pj.head(10)
```

Out[92]:

		Area	Production

Out[92]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
0	Punjab	AMRITSAR	Arhar/Tur	2001	Kharif	1400.0	Hectare	1100.0	Tonnes	0.785714
1	Punjab	AMRITSAR	Arhar/Tur	2002	Kharif	1200.0	Hectare	1000.0	Tonnes	0.833333
2	Punjab	AMRITSAR	Arhar/Tur	2003	Kharif	1500.0	Hectare	1400.0	Tonnes	0.933333
3	Punjab	BATHINDA	Arhar/Tur	2003	Kharif	100.0	Hectare	100.0	Tonnes	1.000000
4	Punjab	FARIDKOT	Arhar/Tur	2001	Kharif	100.0	Hectare	100.0	Tonnes	1.000000
5	Punjab	FARIDKOT	Arhar/Tur	2003	Kharif	300.0	Hectare	300.0	Tonnes	1.000000
6	Punjab	FATEHGARH SAHIB	Arhar/Tur	2001	Kharif	300.0	Hectare	400.0	Tonnes	1.333333
7	Punjab	FATEHGARH SAHIB	Arhar/Tur	2002	Kharif	200.0	Hectare	200.0	Tonnes	1.000000
8	Punjab	FATEHGARH SAHIB	Arhar/Tur	2003	Kharif	200.0	Hectare	200.0	Tonnes	1.000000
9	Punjab	FIROZEPUR	Arhar/Tur	2001	Kharif	200.0	Hectare	200.0	Tonnes	1.000000

In [93]:

```
##production_all = df.groupby(['State', 'Crop', 'Year', 'Season']).sum()[['Production']].res

pj_production = pj.groupby(['Crop', 'Year']).sum()[['Production']].reset_index()
pj_production.head(10)
```

Out[93]:

	Crop	Year	Production
0	Arhar/Tur	1997	8200.0
1	Arhar/Tur	1998	5400.0
2	Arhar/Tur	1999	7200.0
3	Arhar/Tur	2000	7600.0
4	Arhar/Tur	2001	7900.0
5	Arhar/Tur	2002	6700.0
6	Arhar/Tur	2003	9000.0
7	Arhar/Tur	2004	7700.0
8	Arhar/Tur	2005	6900.0
9	Arhar/Tur	2006	6700.0

In [94]:

```
pj_production.Crop.unique()
```

Out[94]:

```
array(['Arhar/Tur', 'Bajra', 'Barley', 'Cotton(lint)', 'Gram',
       'Groundnut', 'Guar seed', 'Jowar', 'Linseed', 'Maize', 'Masoor',
       'Moong(Green Gram)', 'Moth', 'Other Rabi pulses',
       'Peas & beans (Pulses)', 'Rapeseed &Mustard', 'Rice', 'Sesamum',
       'Sugarcane', 'Sunflower', 'Urad', 'Wheat', 'other oilseeds'],
      dtype=object)
```

Plotting the line graphs of crops v production over the years

In [95]:

```
##Arhar/Tur
artur = pj_production[pj_production['Crop'] == 'Arhar/Tur']
artur.head(5)
```

Out[95]:

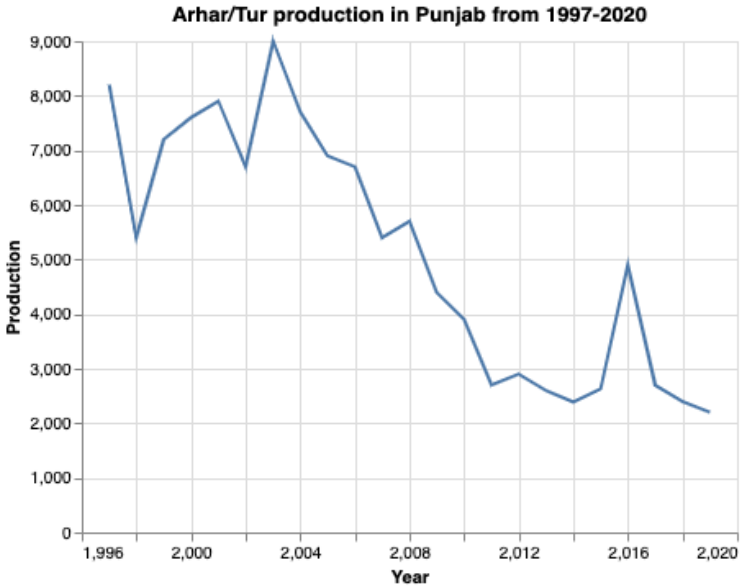
	Crop	Year	Production
0	Arhar/Tur	1997	8200.0
1	Arhar/Tur	1998	5400.0
2	Arhar/Tur	1999	7200.0
3	Arhar/Tur	2000	7600.0

3	Arhar/Tur	2000	7600.0
4	Arhar/Tur	2001	7900.0

In [96]:

```
alt.Chart(artur).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Arhar/Tur production in Punjab from 1997-2020')
```

Out[96]:



There has been a massive decline in arhar/tur crop. This is also called pigeon peas and is a legume.

In [97]:

```
##Barley
barley = pj_production[pj_production['Crop'] == 'Barley']
barley.head(5)
```

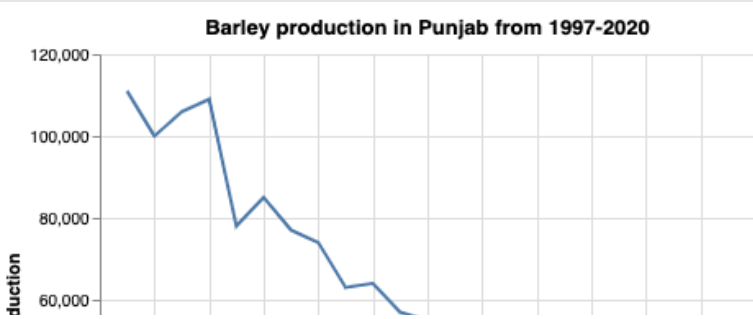
Out[97]:

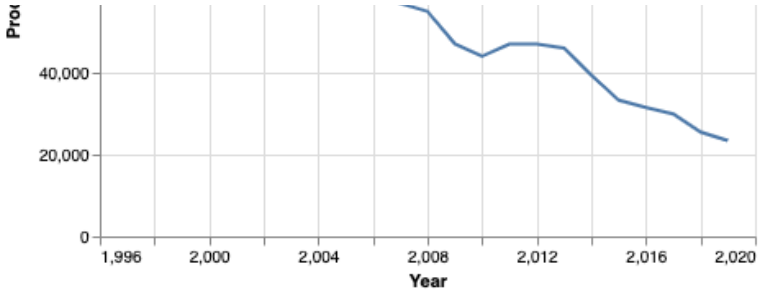
	Crop	Year	Production
43	Barley	1997	111000.0
44	Barley	1998	100000.0
45	Barley	1999	106000.0
46	Barley	2000	109000.0
47	Barley	2001	78000.0

In [98]:

```
alt.Chart(barley).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Barley production in Punjab from 1997-2020')
```

Out[98]:





Similarly with barley.

In [99]:

```
##Cotton(lint)
cotton = pj_production[pj_production['Crop'] == 'Cotton(lint)']
cotton.head(5)
```

Out[99]:

	Crop	Year	Production
66	Cotton(lint)	1997	937000.0
67	Cotton(lint)	1998	595000.0
68	Cotton(lint)	1999	950000.0
69	Cotton(lint)	2000	1199000.0
70	Cotton(lint)	2001	1307000.0

In [100…

```
alt.Chart(cotton).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Cotton(lint) production in Punjab from 1997-2020')
```

Out[100…

In [101…

```
##Gram
gram = pj_production[pj_production['Crop'] == 'Gram']
gram.head(5)
```

Out[101…

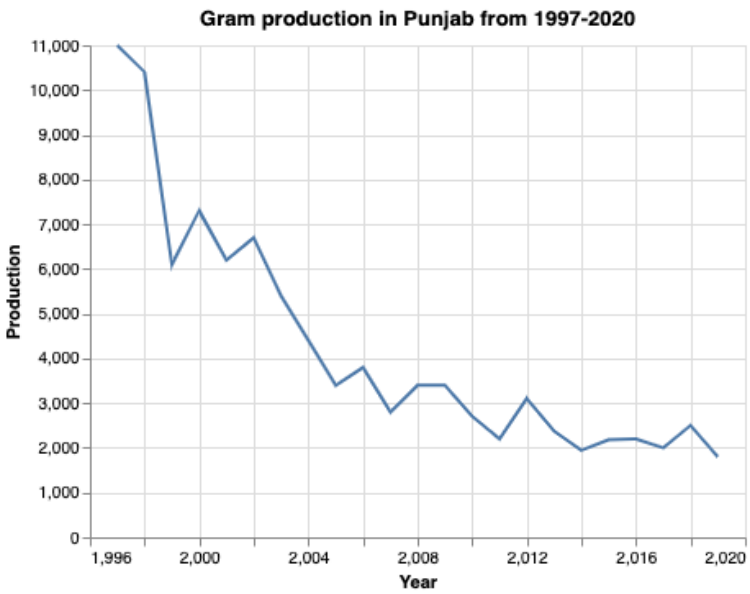
	Crop	Year	Production
89	Gram	1997	11000.0

90	Gram	1998	10400.0
91	Gram	1999	6100.0
92	Gram	2000	7300.0
93	Gram	2001	6200.0

In [102…

```
alt.Chart(gram).mark_line().encode(
    x='Year',
    y=('Production')

).properties(
    title='Gram production in Punjab from 1997-2020')
```

Out[102…



In [103…

```
##Groundnut
gnut = pj_production[pj_production['Crop'] == 'Groundnut']
gnut.head(5)
```

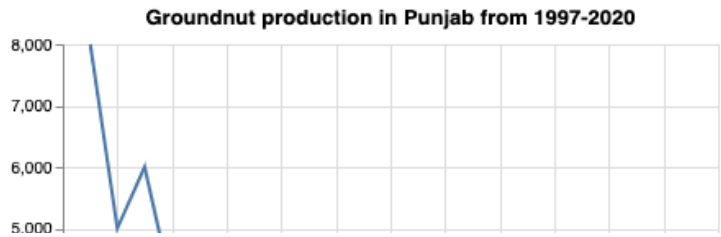
Out[103…

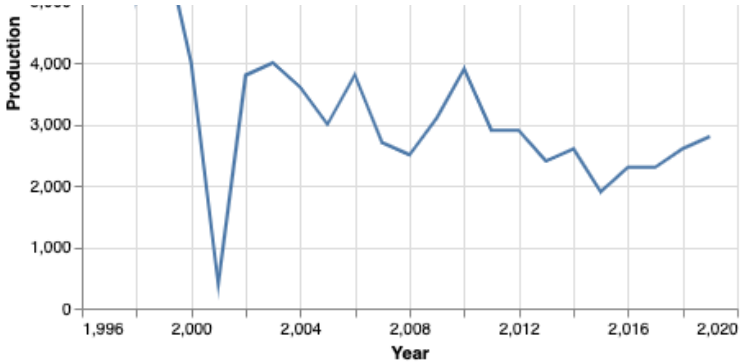
	Crop	Year	Production
112	Groundnut	1997	8000.0
113	Groundnut	1998	5000.0
114	Groundnut	1999	6000.0
115	Groundnut	2000	4000.0
116	Groundnut	2001	400.0

In [104…

```
alt.Chart(gnut).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Groundnut production in Punjab from 1997-2020')
```

Out[104…





In [105…

```
##Guar seeds
guar = pj_production[pj_production['Crop'] == 'Guar seed']
guar.head(5)
```

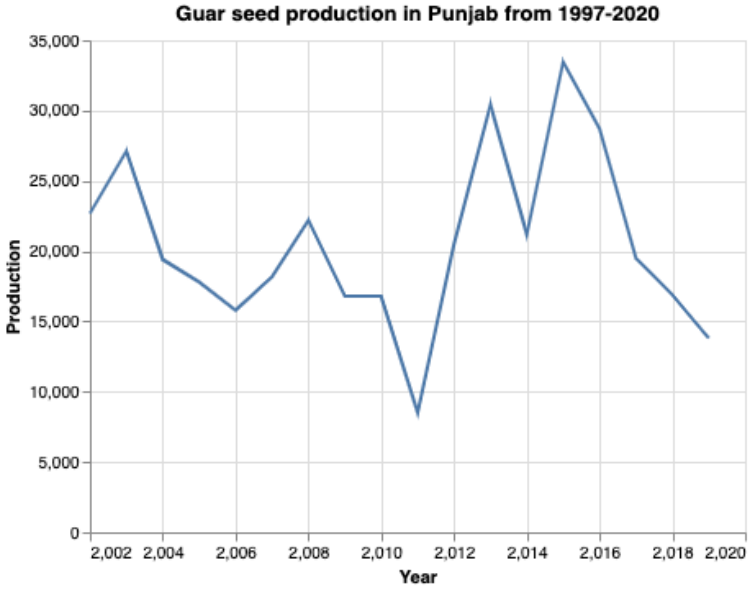
Out[105…

	Crop	Year	Production
135	Guar seed	2002	22700.0
136	Guar seed	2003	27100.0
137	Guar seed	2004	19400.0
138	Guar seed	2005	17800.0
139	Guar seed	2006	15800.0

In [106…

```
alt.Chart(guar).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Guar seed production in Punjab from 1997-2020')
```

Out[106…



In [107…

```
##Jowar
jowar = pj_production[pj_production['Crop'] == 'Jowar']
jowar.head(20)
```

Out[107…

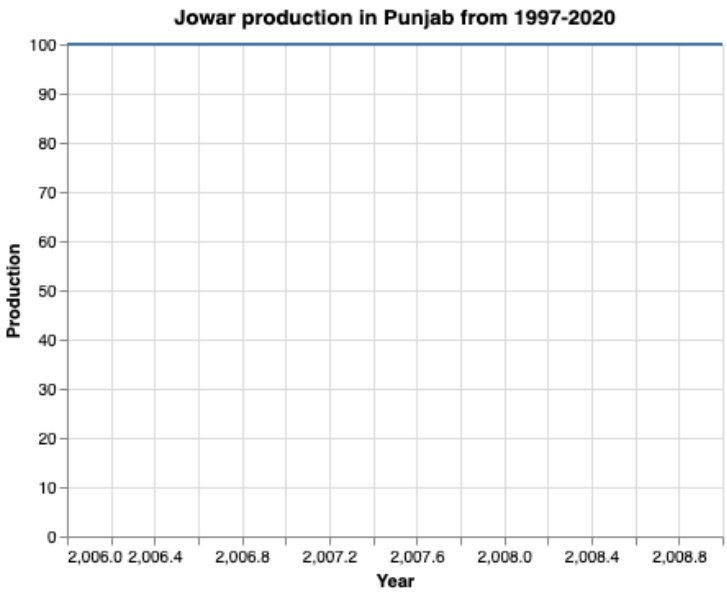
	Crop	Year	Production
153	Jowar	2006	100.0

154	Jowar	2007	100.0
155	Jowar	2008	100.0
156	Jowar	2009	100.0

In [108…

```
alt.Chart(jowar).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Jowar production in Punjab from 1997-2020')
```

Out[108…



Jowar data seems flawed. After checking, it seems like jowar was produced only from 2006-2007, and 100 tonnes each time.

In [109…

```
##Linseed
linseed = pj_production[pj_production['Crop'] == 'Linseed']
linseed.head(5)
```

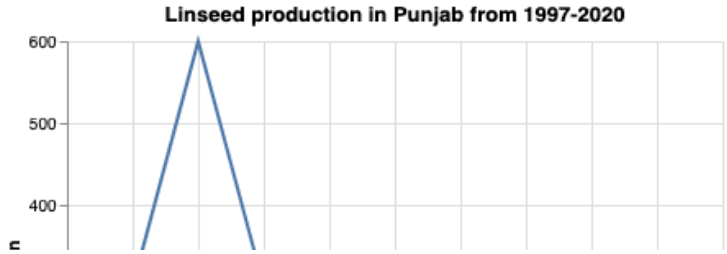
Out[109…

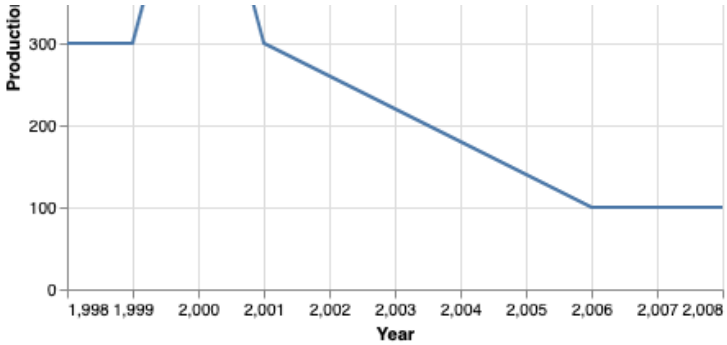
	Crop	Year	Production
157	Linseed	1998	300.0
158	Linseed	1999	300.0
159	Linseed	2000	600.0
160	Linseed	2001	300.0
161	Linseed	2006	100.0

In [110…

```
alt.Chart(linseed).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Linseed production in Punjab from 1997-2020')
```

Out[110…





Linseed prodcuction ended in 2006.

In [111…

```
##Maize
maize = pj_production[pj_production['Crop'] == 'Maize']
maize.head(5)
```

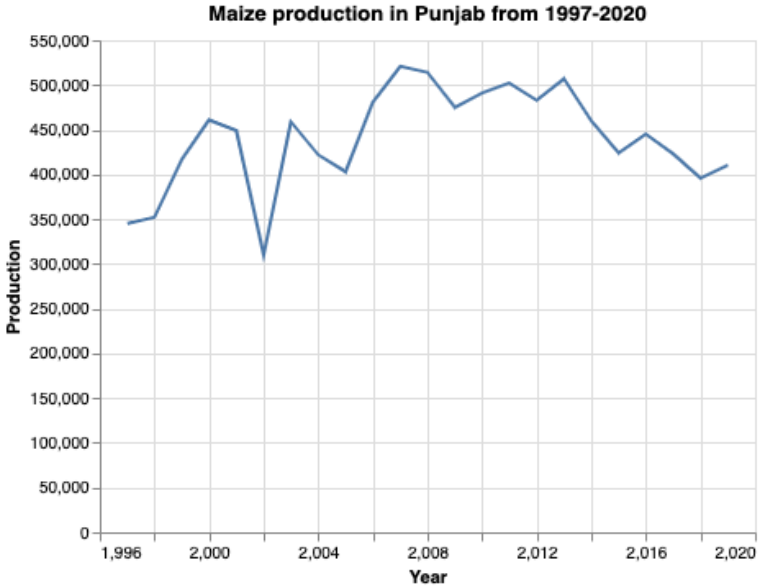
Out[111…

	Crop	Year	Production
163	Maize	1997	345000.0
164	Maize	1998	352000.0
165	Maize	1999	417000.0
166	Maize	2000	461000.0
167	Maize	2001	449000.0

In [112…

```
alt.Chart(maize).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Maize production in Punjab from 1997-2020')
```

Out[112…



In [113…

```
#Masoor
masoor = pj_production[pj_production['Crop'] == 'Masoor']
masoor.head(5)
```

Out[113…

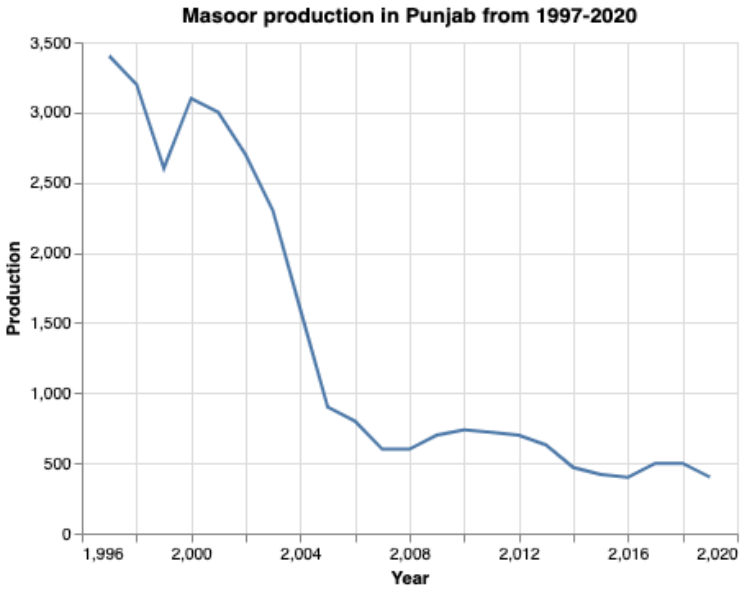
	Crop	Year	Production

186	Masoor	1997	3400.0
187	Masoor	1998	3200.0
188	Masoor	1999	2600.0
189	Masoor	2000	3100.0
190	Masoor	2001	3000.0

In [114…

```
alt.Chart(masoor).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Masoor production in Punjab from 1997-2020')
```

Out[114…



Masoor is a dal or a legume crop.

In [115…

```
moong = pj_production[pj_production['Crop'] == 'Moong(Green Gram)']
moong.head(5)
```

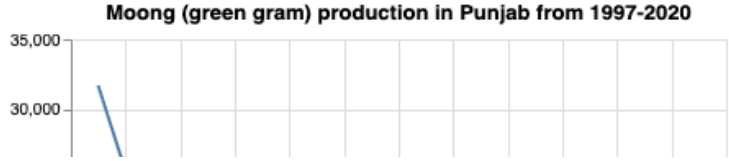
Out[115…

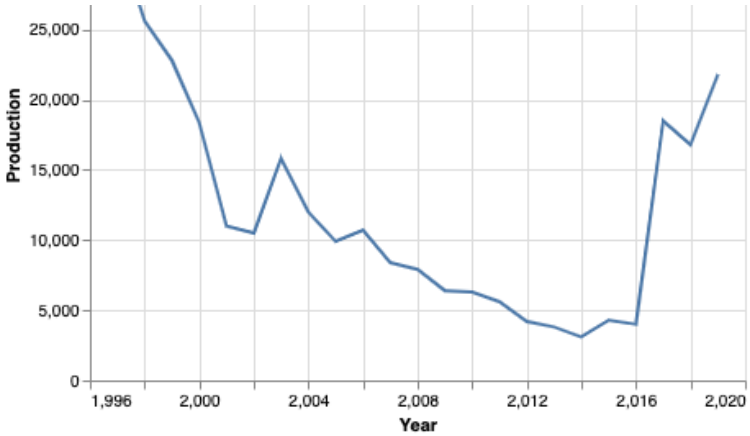
	Crop	Year	Production
209	Moong(Green Gram)	1997	31700.0
210	Moong(Green Gram)	1998	25600.0
211	Moong(Green Gram)	1999	22800.0
212	Moong(Green Gram)	2000	18400.0
213	Moong(Green Gram)	2001	11000.0

In [116…

```
alt.Chart(moong).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Moong (green gram) production in Punjab from 1997-2020')
```

Out[116…





Moong production seems to be reviving.

In [117…

```
#Moth
moth = pj_production[pj_production['Crop'] == 'Moth']
moth.head(20)
```

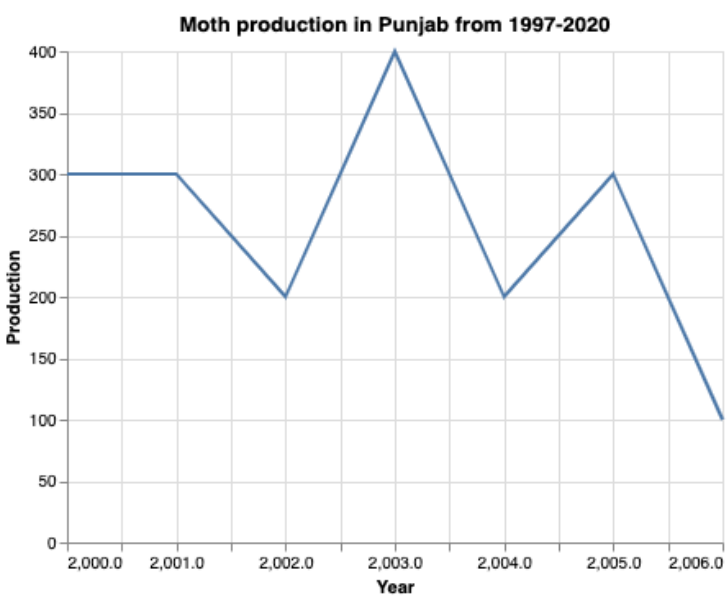
Out[117…

	Crop	Year	Production
232	Moth	2000	300.0
233	Moth	2001	300.0
234	Moth	2002	200.0
235	Moth	2003	400.0
236	Moth	2004	200.0
237	Moth	2005	300.0
238	Moth	2006	100.0

In [118…

```
alt.Chart(moth).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Moth production in Punjab from 1997-2020')
```

Out[118…



Moth was grown only from 2000-2006.

In [119…

```
#Other Rabi Pulses - rabi is the winter cropping season
rabi = pj_production[pj_production['Crop'] == 'Other Rabi pulses']
rabi.head(5)
```

Out[119…

	Crop	Year	Production
239	Other Rabi pulses	2000	6300.0

Only 2000 year has this dataset recorded.

In [120…

```
#Peas and beans
pb = pj_production[pj_production['Crop'] == 'Peas & beans (Pulses)']
pb.head(5)
```

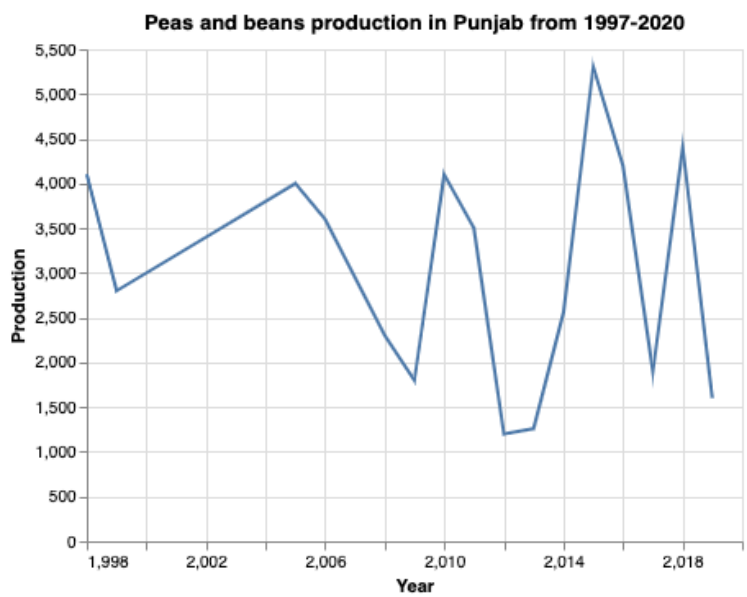
Out[120…

	Crop	Year	Production
240	Peas & beans (Pulses)	1998	4100.0
241	Peas & beans (Pulses)	1999	2800.0
242	Peas & beans (Pulses)	2005	4000.0
243	Peas & beans (Pulses)	2006	3600.0
244	Peas & beans (Pulses)	2008	2300.0

In [121…

```
alt.Chart(pb).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Peas and beans production in Punjab from 1997-2020')
```

Out[121…



In [122…

```
#Rapeseed and mustard
rm = pj_production[pj_production['Crop'] == 'Rapeseed &Mustard']
rm.head(5)
```

Out[122…

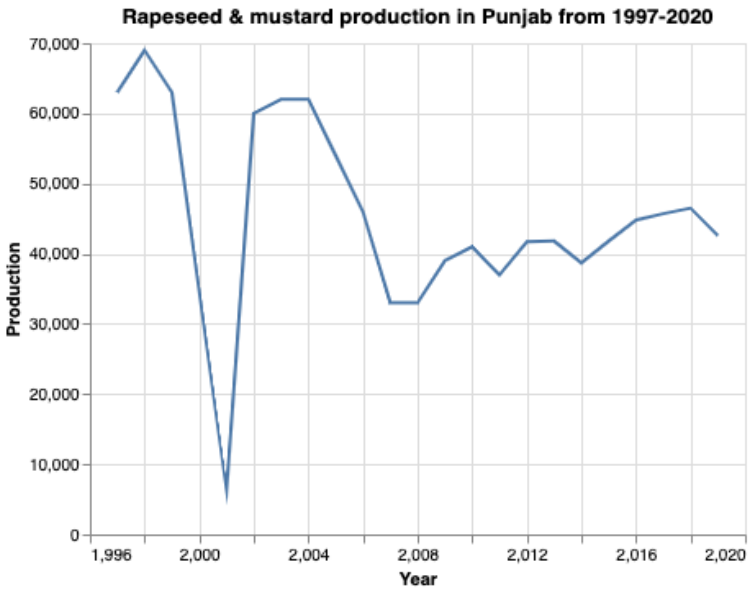
	Crop	Year	Production
256	Rapeseed &Mustard	1997	63000.0
257	Rapeseed &Mustard	1998	69000.0

258	Rapeseed &Mustard	1999	63000.0
259	Rapeseed &Mustard	2001	6500.0
260	Rapeseed &Mustard	2002	60000.0

In [123…

```
alt.Chart(rm).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Rapeseed & mustard production in Punjab from 1997-2020')
```

Out[123…



In [124…

```
#Sesamum
ses = pj_production[pj_production['Crop'] == 'Sesamum']
ses.head(5)
```

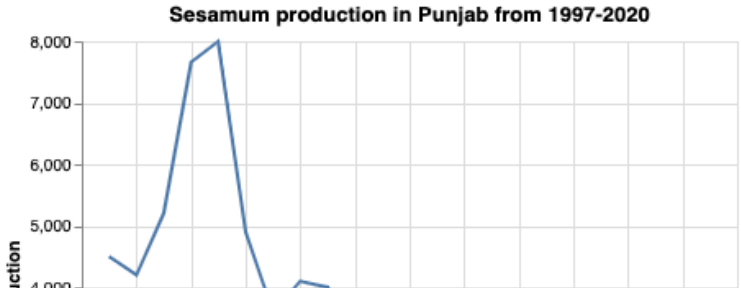
Out[124…

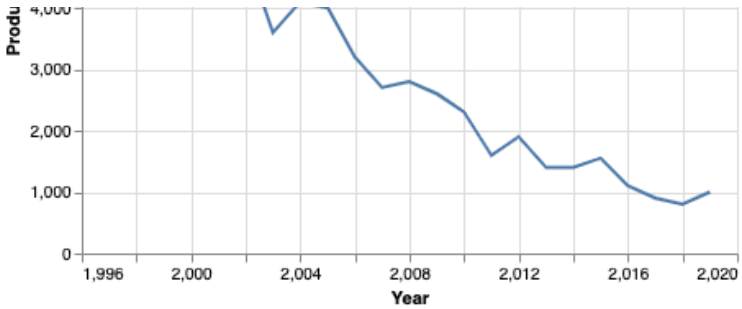
	Crop	Year	Production
301	Sesamum	1997	4500.0
302	Sesamum	1998	4200.0
303	Sesamum	1999	5200.0
304	Sesamum	2000	7660.0
305	Sesamum	2001	8000.0

In [125…

```
alt.Chart(ses).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Sesamum production in Punjab from 1997-2020')
```

Out[125…





In [126…

```
##Sugarcane
sugarcane = pj_production[pj_production['Crop'] == 'Sugarcane']
sugarcane.head(5)
```

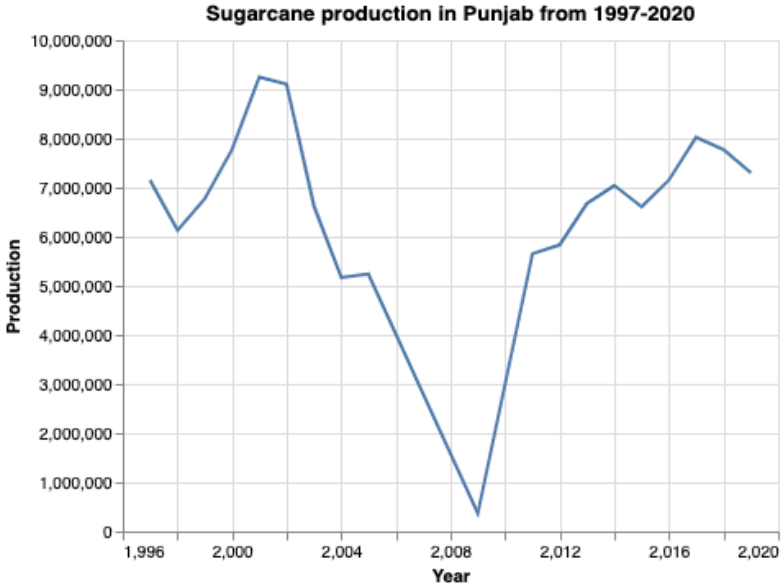
Out[126…

	Crop	Year	Production
324	Sugarcane	1997	7150000.0
325	Sugarcane	1998	6130000.0
326	Sugarcane	1999	6770000.0
327	Sugarcane	2000	7770000.0
328	Sugarcane	2001	9250000.0

In [127…

```
alt.Chart(sugarcane).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Sugarcane production in Punjab from 1997-2020')
```

Out[127…



In [128…

```
##Sunflower
sunflower = pj_production[pj_production['Crop'] == 'Sunflower']
sunflower.head(5)
```

Out[128…

	Crop	Year	Production
343	Sunflower	2003	34000.0
344	Sunflower	2012	23700.0

345	Sunflower	2016	9600.0
346	Sunflower	2017	11600.0
347	Sunflower	2018	9700.0

This is an incomplete dataset.

In [129…

```
alt.Chart(sunflower).mark_line().encode(
    x='Year',
    y=('Production')
).properties(
    title='Sunflower production in Punjab from 1997-2020')
```

Out[129…