Задание. Создать ноутбук, который содержит следующие разделы: Текстовое описание выбранного Вами набора данных. Основные характеристики датасета. Визуальное исследование датасета. Информация о корреляции признаков.

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [2]:

```
data = pd.read_csv('C:/Users/maxim/OneDrive/Рабочий стол/archive/rice_wheat_corn_prices.csv', sep=',')
```

In [3]:

```
data.head()
```

Out[3]:

	Year	Month	Price_wheat_ton	Price_rice_ton	Price_corn_ton	Inflation_rate	Price_wheat_ton_infl	Price_rice_ton_infl	Price_corn_ton_infl
0	1992	Feb	170.12	278.25	113.62	89.59	322.53	527.53	215.41
1	1992	Mar	161.44	277.20	117.00	89.59	306.07	525.54	221.82
2	1992	Apr	153.07	278.00	108.52	89.59	290.21	527.06	205.74
3	1992	May	139.72	274.00	109.64	89.59	264.90	519.48	207.87
4	1992	Jun	140.36	268.80	110.90	89.59	266.11	509.62	210.26

In [4]:

```
data.shape
```

Out[4]:

```
(359, 9)
```

In [5]:

```
data.columns
```

Out[5]:

```
Index(['Year', 'Month', 'Price_wheat_ton', 'Price_rice_ton', 'Price_corn_ton',
       'Inflation_rate', 'Price_wheat_ton_infl', 'Price_rice_ton_infl',
       'Price_corn_ton_infl'],
      dtype='object')
```

In [6]:

```
data.dtypes
```

Out[6]:

```
Year                   int64
Month                 object
Price_wheat_ton      float64
Price_rice_ton       float64
Price_corn_ton       float64
Inflation_rate       float64
Price_wheat_ton_infl float64
Price_rice_ton_infl  float64
Price_corn_ton_infl  float64
dtype: object
```

In [7]:

```
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
Year - 0
Month - 0
Price_wheat_ton - 0
Price_rice_ton - 0
Price_corn_ton - 0
Inflation_rate - 0
Price_wheat_ton_infl - 0
Price_rice_ton_infl - 0
Price_corn_ton_infl - 0
```

In [8]:

```
data.describe()
```

Out[8]:

	Year	Price_wheat_ton	Price_rice_ton	Price_corn_ton	Inflation_rate	Price_wheat_ton_infl	Price_rice_ton_infl	Price_corn_ton_infl
count	359.000000	359.000000	359.000000	359.000000	359.000000	359.000000	359.000000	359.000000
mean	2006.540390	185.302869	363.930418	155.165376	36.316685	241.726769	474.038384	201.211086
std	8.645592	64.985279	131.508817	62.370286	26.605378	65.234300	133.776144	61.727752
min	1992.000000	85.300000	163.750000	75.270000	-1.290000	136.220000	246.020000	116.280000
25%	1999.000000	137.315000	261.250000	104.155000	12.370000	193.010000	397.500000	159.015000
50%	2007.000000	175.270000	343.750000	149.340000	28.300000	228.310000	455.540000	186.020000
75%	2014.000000	220.260000	444.500000	176.010000	59.700000	275.835000	540.180000	220.940000
max	2021.000000	419.610000	907.000000	333.050000	89.590000	518.470000	1120.690000	385.910000

In [9]:

```
data['Year'].unique()
```

Out[9]:

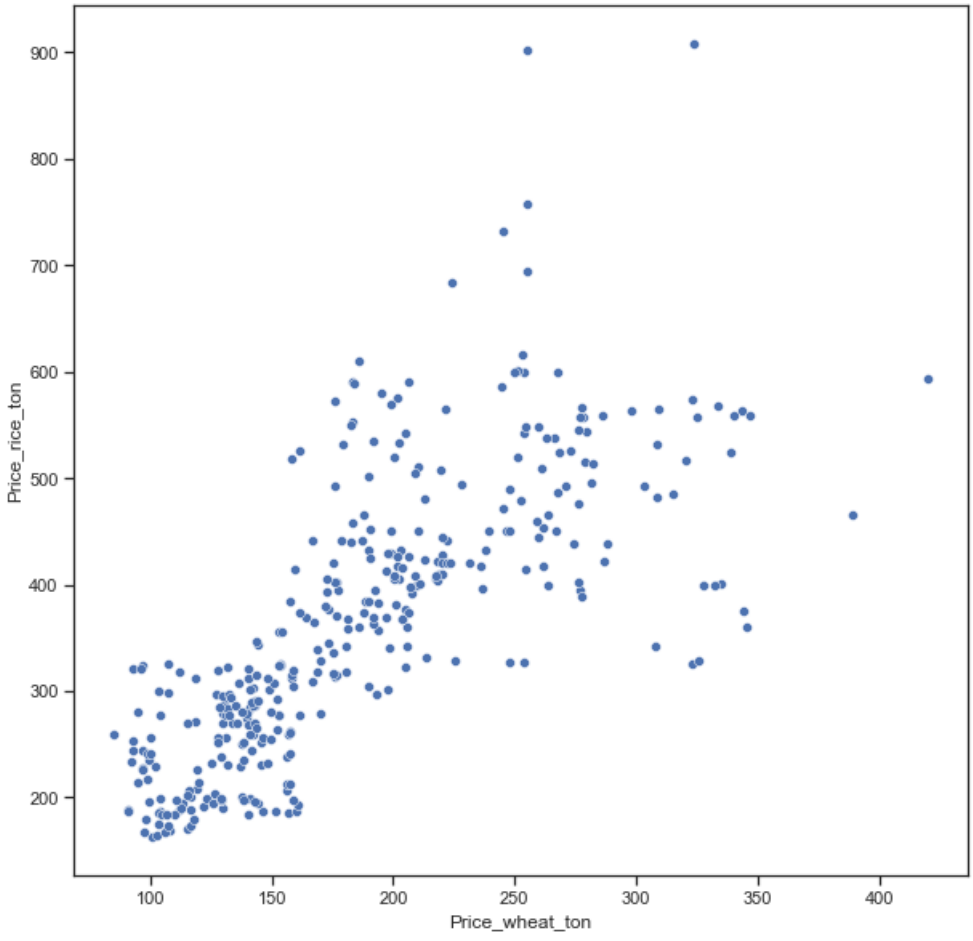
```
array([1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
       2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013,
       2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021], dtype=int64)
```

In [11]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Price_wheat_ton', y='Price_rice_ton', data=data)
```

Out[11]:

```
<AxesSubplot:xlabel='Price_wheat_ton', ylabel='Price_rice_ton'>
```

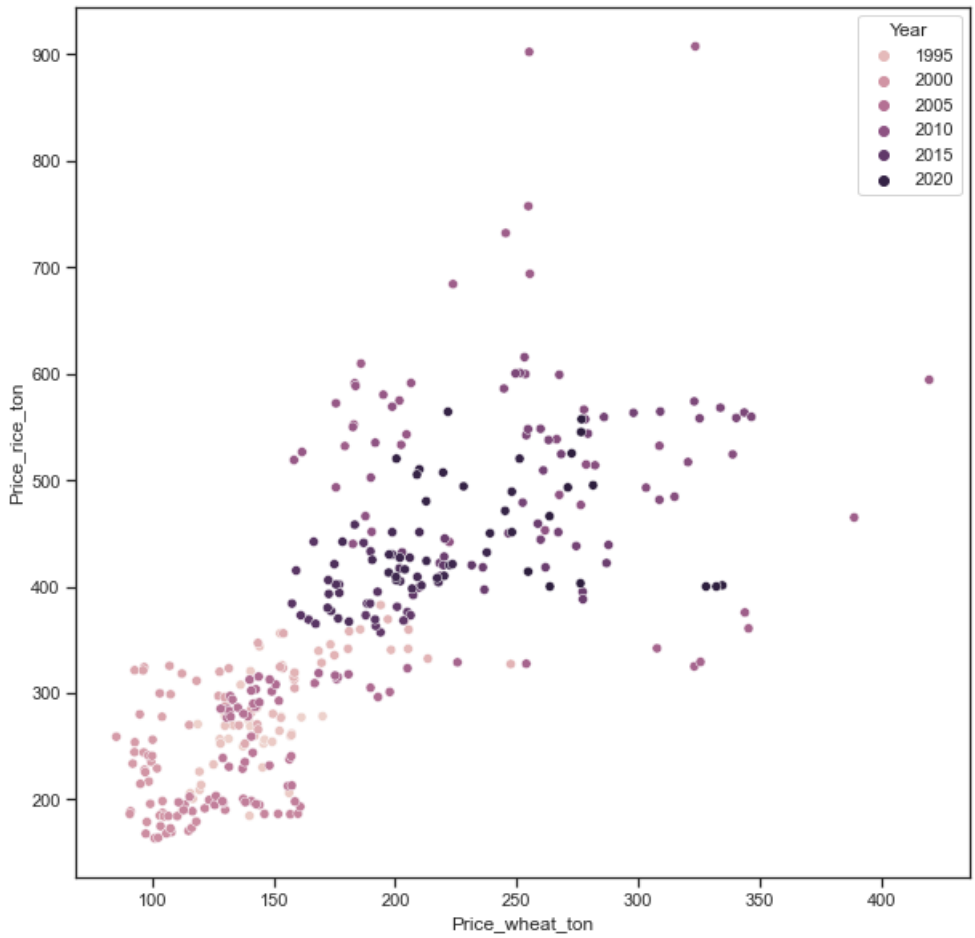


In [12]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Price_wheat_ton', y='Price_rice_ton', data=data, hue='Year')
```

Out[12]:

<AxesSubplot:xlabel='Price_wheat_ton', ylabel='Price_rice_ton'>



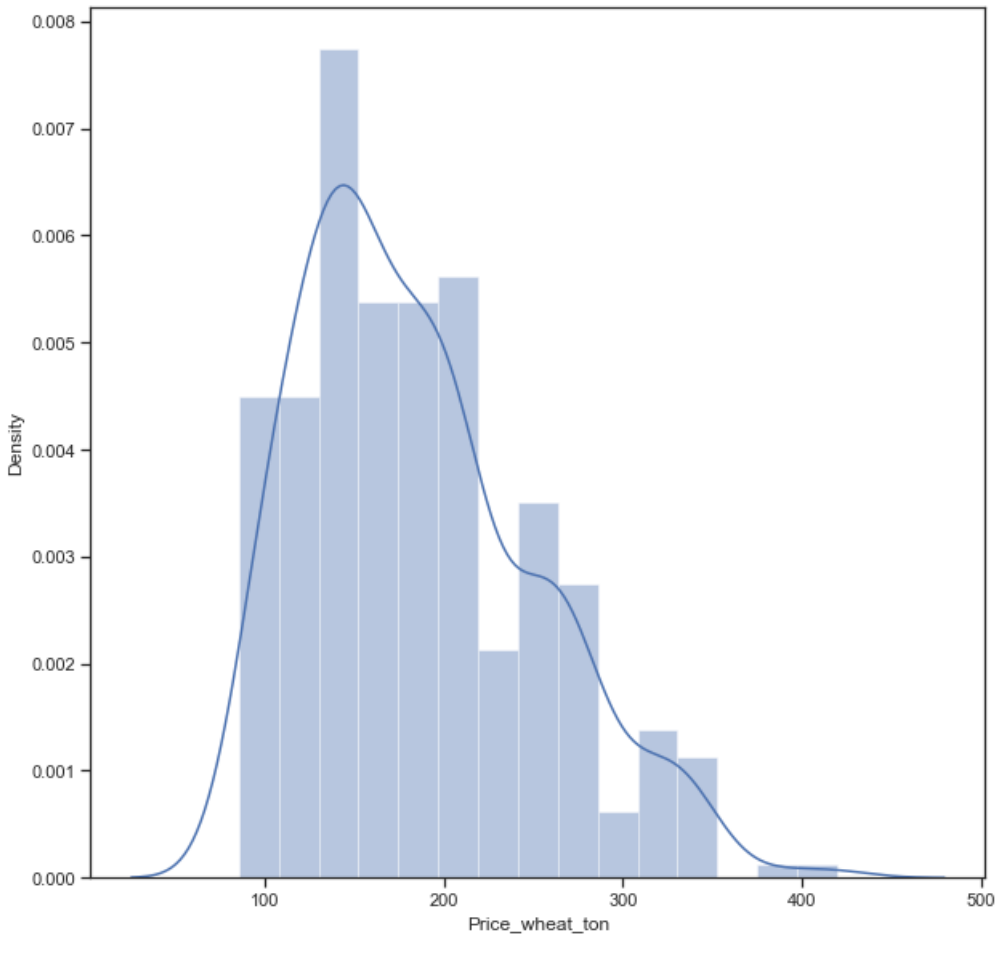
In [14]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Price_wheat_ton'])
```

C:\Users\maxim\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

Out[14]:

<AxesSubplot:xlabel='Price_wheat_ton', ylabel='Density'>

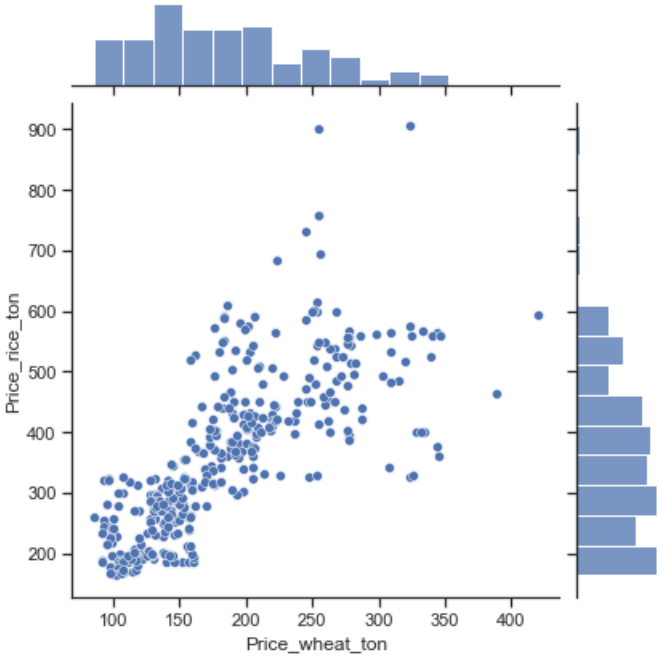


In [15]:

```
sns.jointplot(x='Price_wheat_ton', y='Price_rice_ton', data=data)
```

Out[15]:

```
<seaborn.axisgrid.JointGrid at 0x2a49e2fc9d0>
```

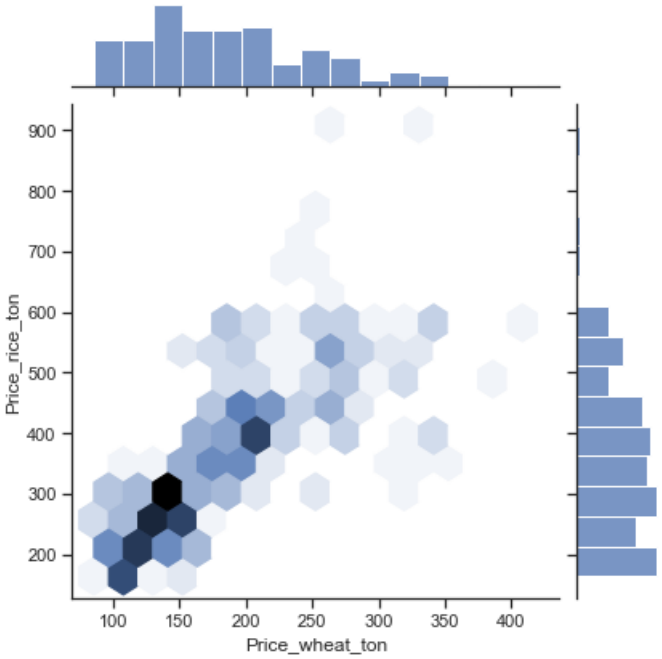


In [16]:

```
sns.jointplot(x='Price_wheat_ton', y='Price_rice_ton', data=data, kind="hex")
```

Out[16]:

```
<seaborn.axisgrid.JointGrid at 0x2a49e3cf970>
```

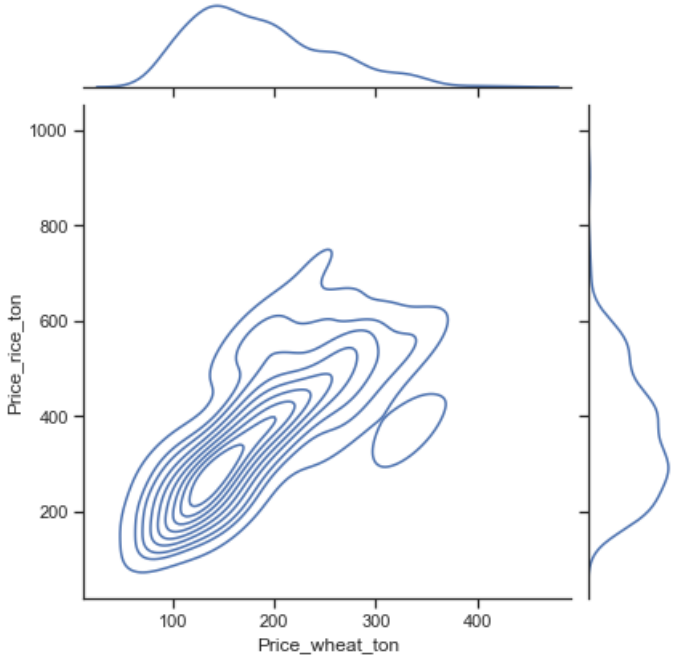


In [17]:

```
sns.jointplot(x='Price_wheat_ton', y='Price_rice_ton', data=data, kind="kde")
```

Out[17]:

<seaborn.axisgrid.JointGrid at 0x2a49e6f0910>

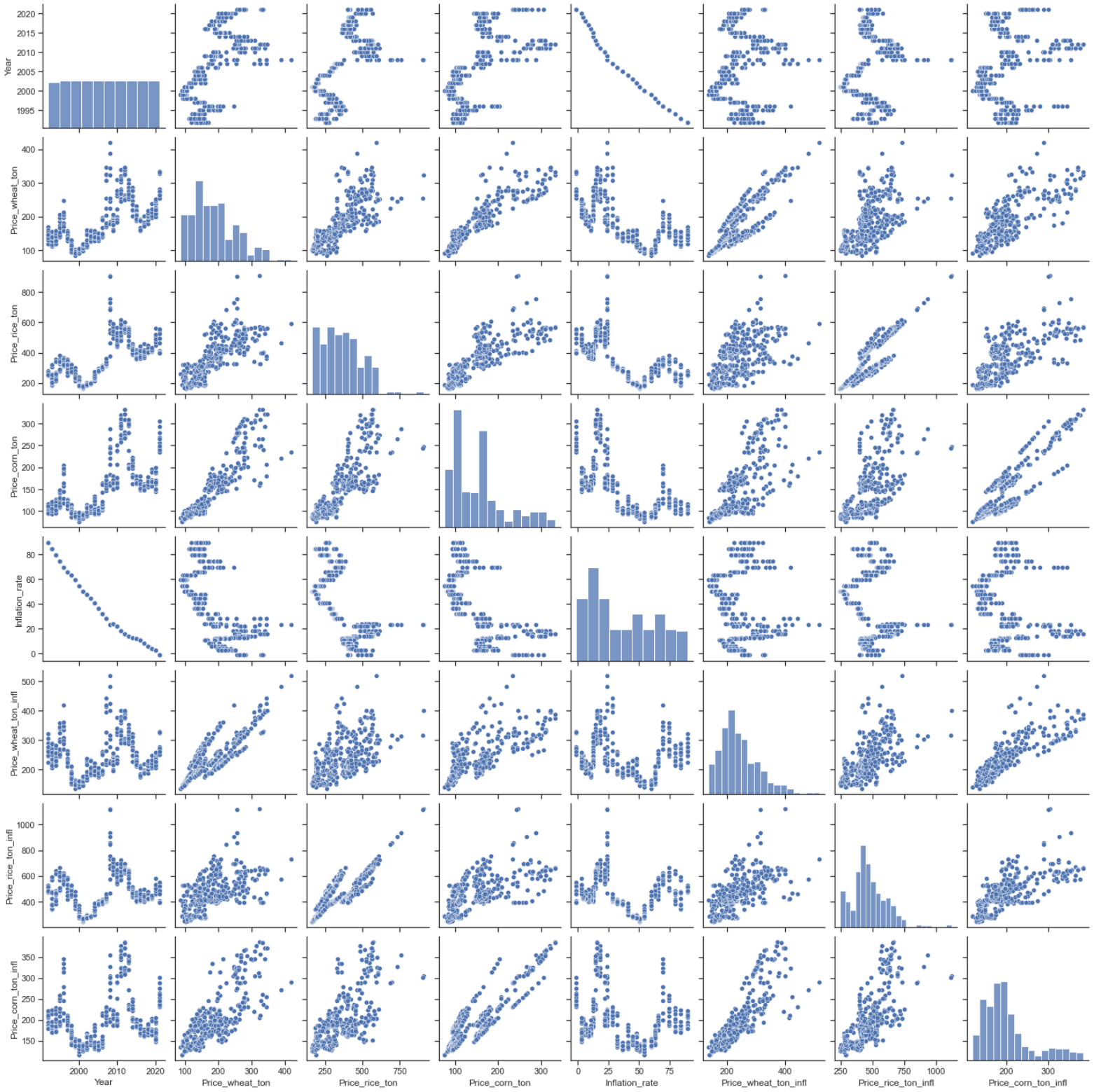


In [19]:

```
sns.pairplot(data)
```

Out[19]:

<seaborn.axisgrid.PairGrid at 0x2a4a17e6790>

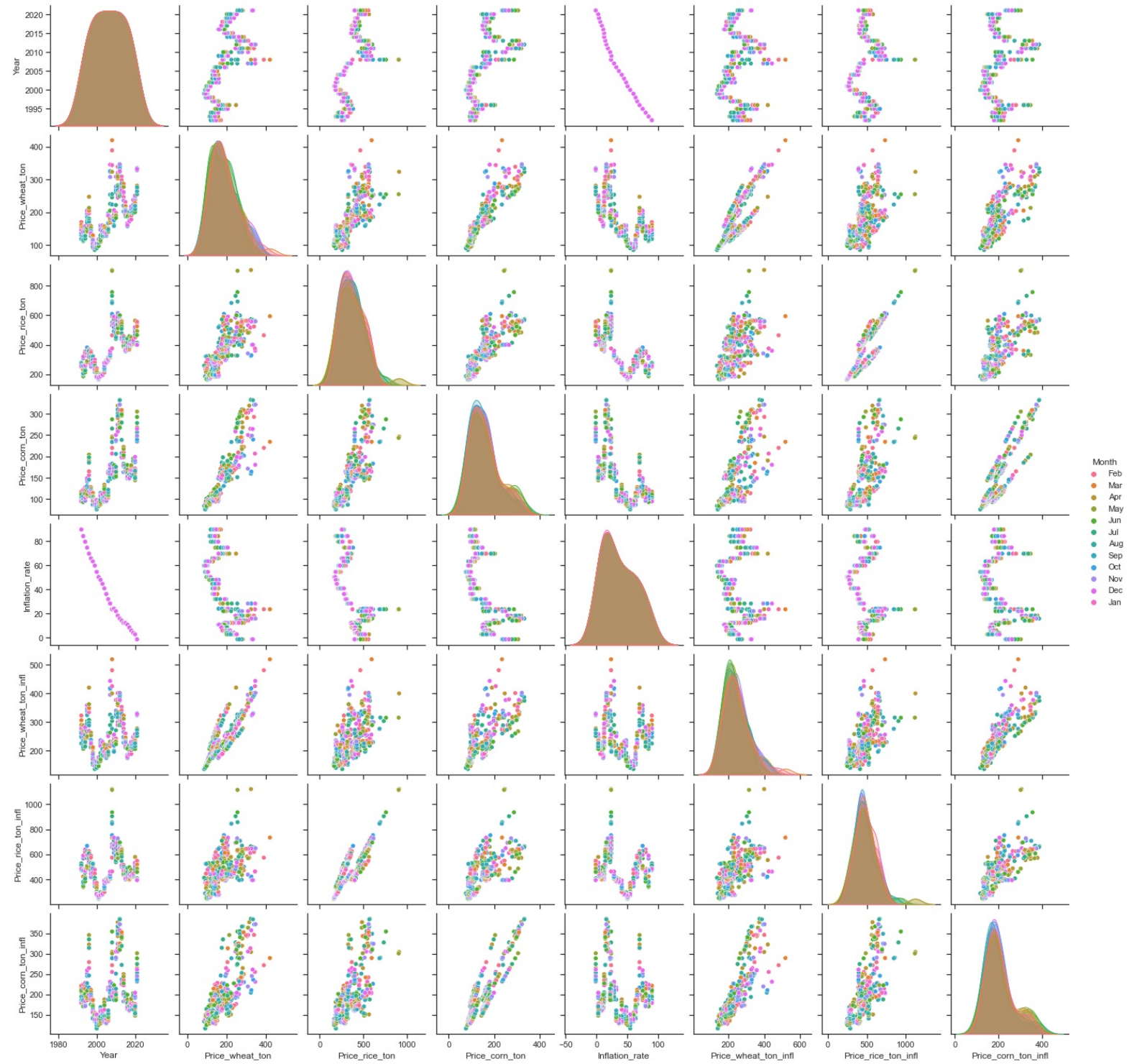


In [21]:

```
sns.pairplot(data, hue="Month")
```

Out[21]:

<seaborn.axisgrid.PairGrid at 0x2a4a4e02ac0>

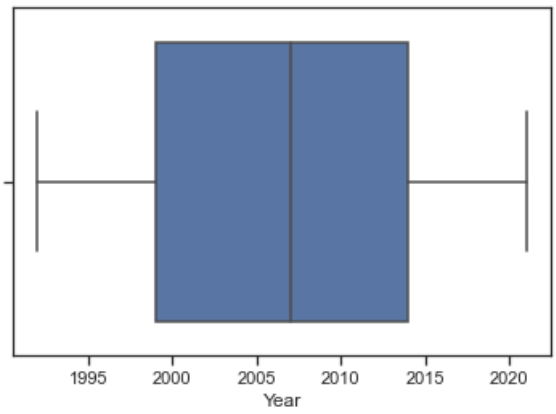


In [22]:

```
sns.boxplot(x=data['Year'])
```

Out[22]:

<AxesSubplot:xlabel='Year'>

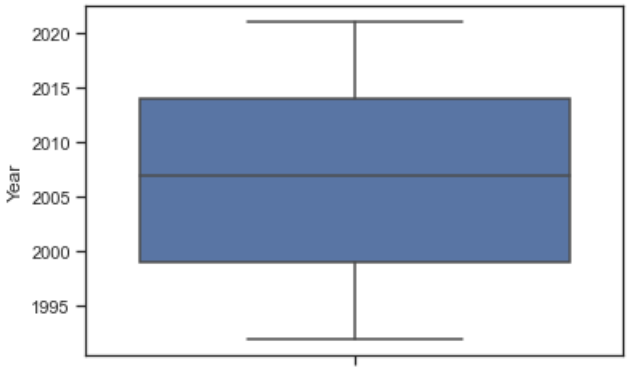


In [23]:

```
sns.boxplot(y=data['Year'])
```

Out[23]:

<AxesSubplot:ylabel='Year'>

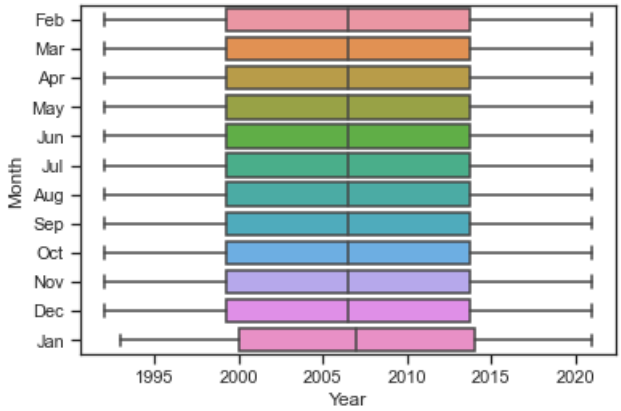


In [24]:

```
sns.boxplot(x='Year', y='Month', data=data)
```

Out[24]:

<AxesSubplot:xlabel='Year', ylabel='Month'>

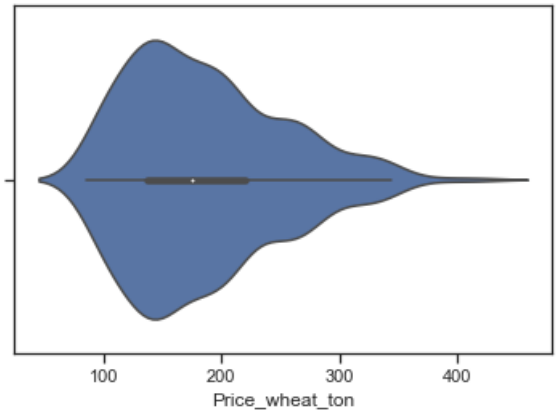


In [27]:

```
sns.violinplot(x=data['Price_wheat_ton'])
```

Out[27]:

<AxesSubplot:xlabel='Price_wheat_ton'>



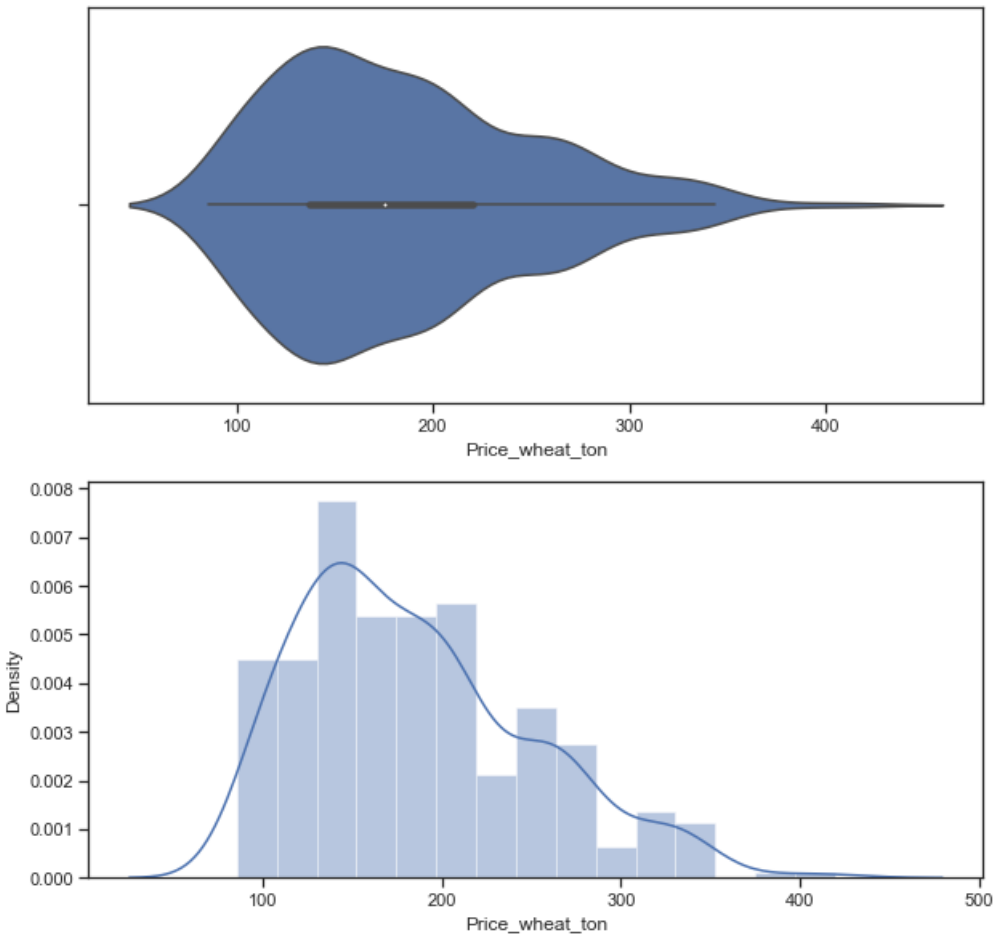
In [28]:

```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['Price_wheat_ton'])
sns.distplot(data['Price_wheat_ton'], ax=ax[1])
```

```
C:\Users\maxim\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[28]:

```
<AxesSubplot:xlabel='Price_wheat_ton', ylabel='Density'>
```

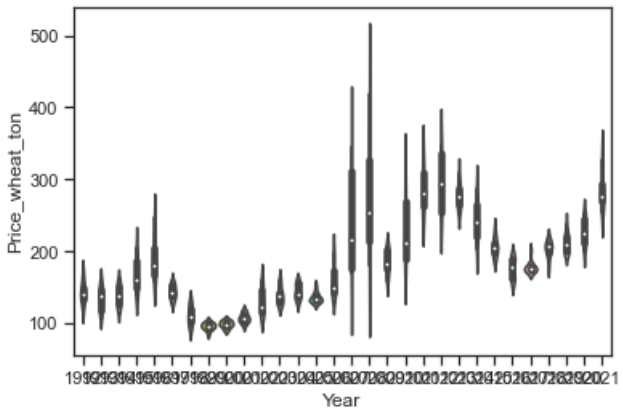


In [30]:

```
sns.violinplot(x='Year', y='Price_wheat_ton', data=data)
```

Out[30]:

```
<AxesSubplot:xlabel='Year', ylabel='Price_wheat_ton'>
```

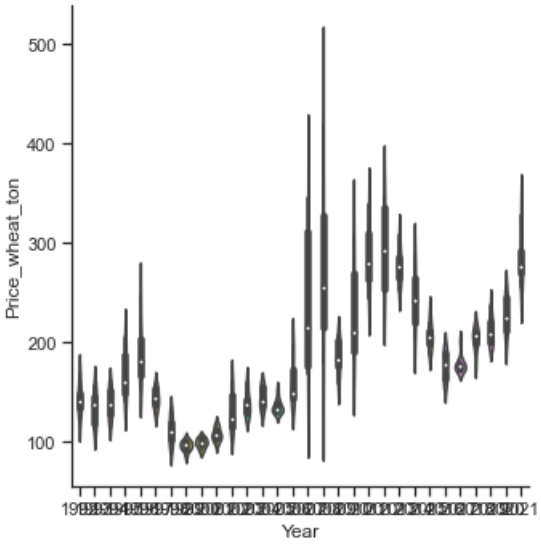


In [32]:

```
sns.catplot(y='Price_wheat_ton', x='Year', data=data, kind="violin", split=True)
```

Out[32]:

<seaborn.axisgrid.FacetGrid at 0x2a4aabb9a90>



In [33]:

```
data.corr()
```

Out[33]:

	Year	Price_wheat_ton	Price_rice_ton	Price_corn_ton	Inflation_rate	Price_wheat_ton_infl	Price_rice_ton_infl	Price_corn_ton_infl
Year	1.000000	0.614216	0.612864	0.601719	-0.985230	0.102244	0.129089	0.190547
Price_wheat_ton	0.614216	1.000000	0.747221	0.892489	-0.630568	0.838342	0.536606	0.762467
Price_rice_ton	0.612864	0.747221	1.000000	0.808799	-0.632244	0.528453	0.856352	0.669503
Price_corn_ton	0.601719	0.892489	0.808799	1.000000	-0.622761	0.696853	0.612011	0.890928
Inflation_rate	-0.985230	-0.630568	-0.632244	-0.622761	1.000000	-0.121858	-0.152360	-0.217992
Price_wheat_ton_infl	0.102244	0.838342	0.528453	0.696853	-0.121858	1.000000	0.600255	0.821319
Price_rice_ton_infl	0.129089	0.536606	0.856352	0.612011	-0.152360	0.600255	1.000000	0.707511
Price_corn_ton_infl	0.190547	0.762467	0.669503	0.890928	-0.217992	0.821319	0.707511	1.000000

In [34]:

```
data.corr(method='pearson')
```

Out[34]:

	Year	Price_wheat_ton	Price_rice_ton	Price_corn_ton	Inflation_rate	Price_wheat_ton_infl	Price_rice_ton_infl	Price_corn_ton_infl
Year	1.000000	0.614216	0.612864	0.601719	-0.985230	0.102244	0.129089	0.190547
Price_wheat_ton	0.614216	1.000000	0.747221	0.892489	-0.630568	0.838342	0.536606	0.762467
Price_rice_ton	0.612864	0.747221	1.000000	0.808799	-0.632244	0.528453	0.856352	0.669503
Price_corn_ton	0.601719	0.892489	0.808799	1.000000	-0.622761	0.696853	0.612011	0.890928
Inflation_rate	-0.985230	-0.630568	-0.632244	-0.622761	1.000000	-0.121858	-0.152360	-0.217992
Price_wheat_ton_infl	0.102244	0.838342	0.528453	0.696853	-0.121858	1.000000	0.600255	0.821319
Price_rice_ton_infl	0.129089	0.536606	0.856352	0.612011	-0.152360	0.600255	1.000000	0.707511
Price_corn_ton_infl	0.190547	0.762467	0.669503	0.890928	-0.217992	0.821319	0.707511	1.000000

In [35]:

```
data.corr(method='kendall')
```

Out[35]:

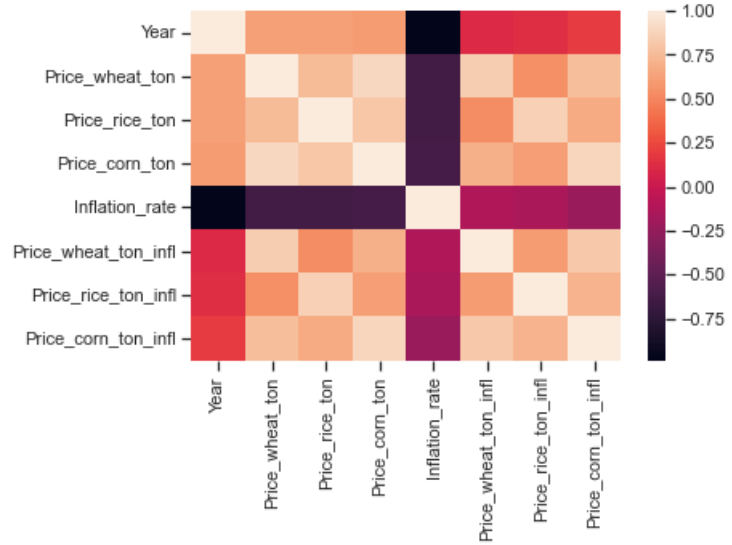
	Year	Price_wheat_ton	Price_rice_ton	Price_corn_ton	Inflation_rate	Price_wheat_ton_infl	Price_rice_ton_infl	Price_corn_ton_infl
Year	1.000000	0.461176	0.423723	0.432980	-0.995377	0.085667	0.080066	0.096277
Price_wheat_ton	0.461176	1.000000	0.621697	0.776169	-0.464463	0.630291	0.403515	0.550994
Price_rice_ton	0.423723	0.621697	1.000000	0.668084	-0.425810	0.389846	0.661939	0.480850
Price_corn_ton	0.432980	0.776169	0.668084	1.000000	-0.436457	0.547066	0.453341	0.668519
Inflation_rate	-0.995377	-0.464463	-0.425810	-0.436457	1.000000	-0.088891	-0.082152	-0.099691
Price_wheat_ton_infl	0.085667	0.630291	0.389846	0.547066	-0.088891	1.000000	0.488561	0.705246
Price_rice_ton_infl	0.080066	0.403515	0.661939	0.453341	-0.082152	0.488561	1.000000	0.556476
Price_corn_ton_infl	0.096277	0.550994	0.480850	0.668519	-0.099691	0.705246	0.556476	1.000000

In [36]:

```
sns.heatmap(data.corr())
```

Out[36]:

<AxesSubplot:>

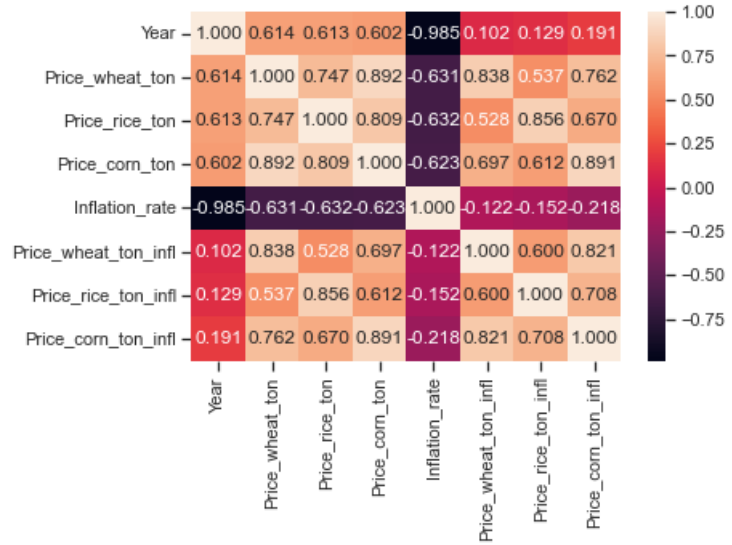


In [37]:

```
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[37]:

<AxesSubplot:>

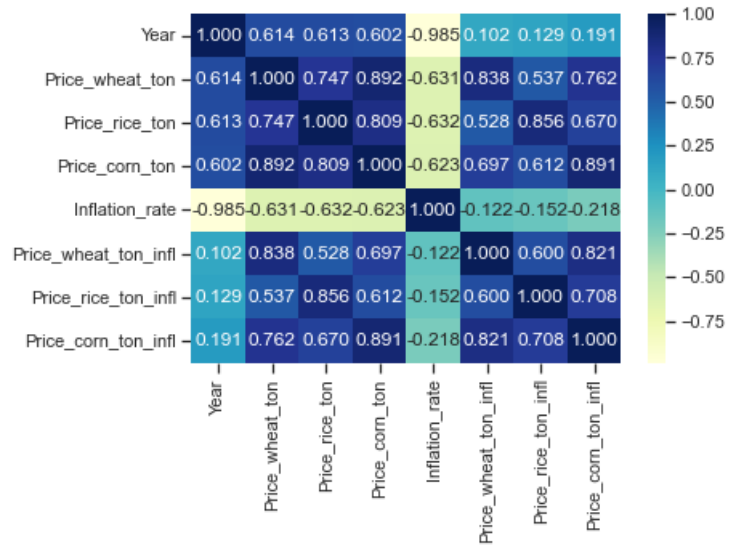


In [38]:

```
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

Out[38]:

<AxesSubplot:>



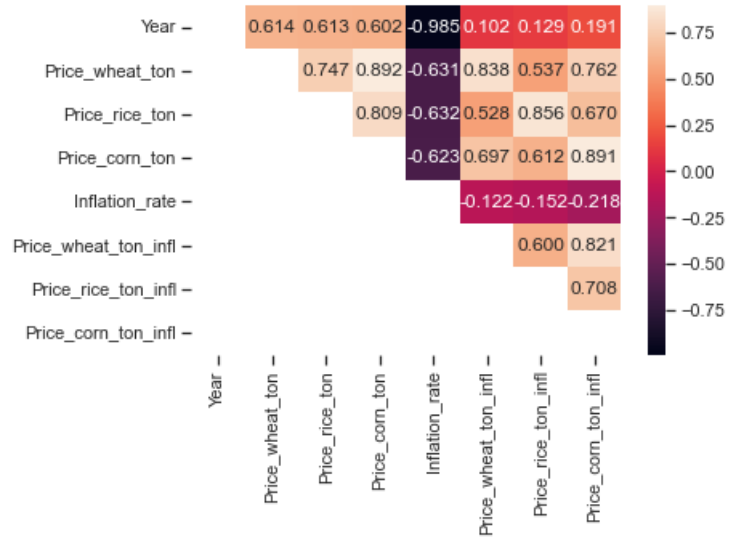
In [39]:

```
mask = np.zeros_like(data.corr(), dtype=np.bool)
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

```
C:\Users\maxim\AppData\Local\Temp/ipykernel_6468/1053490320.py:1: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  mask = np.zeros_like(data.corr(), dtype=np.bool)
```

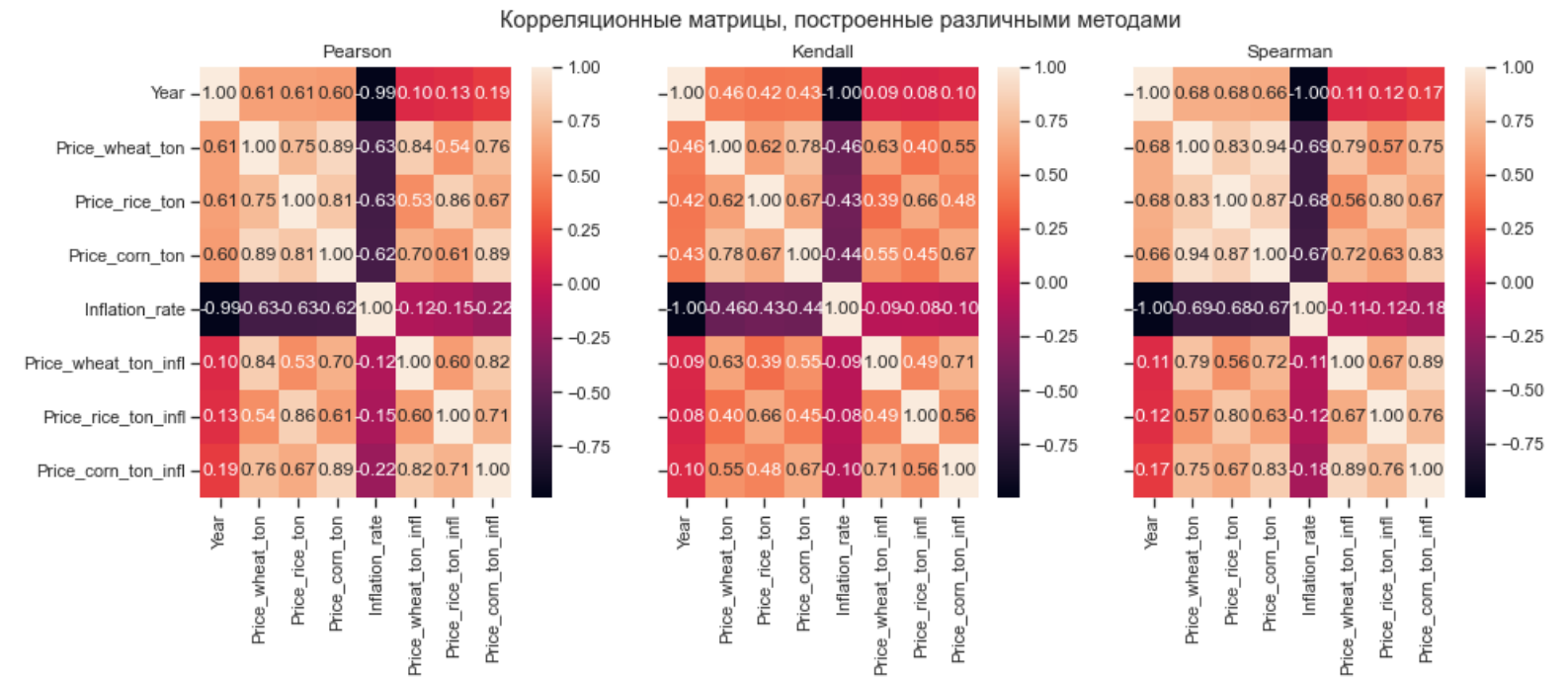
Out[39]:

<AxesSubplot:>



In [40]:

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```


Корреляционные матрицы, построенные различными методами
Pearson
Kendall
Spearman
Year
Price_wheat_ton
Price_rice_ton
Price_corn_ton
Inflation_rate
Price_wheat_ton_infl
Price_rice_ton_infl
Price_corn_ton_infl
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

In [ ]: