



# 美國物價預測

## Commodity Prices Analysis

108590026 劉俊廷  
109590001 林昕昱  
109590020 王郁鈞  
109590021 古佳豫  
109590039 王泉裕

# 動機

—— 做出投資決策

—— 制定財務計劃

—— 企業經營

—— 預防風險



# 美國物價新聞

## 美國2021年消費者物價指數飆升7% 創近40年新高

**yahoo!**國際通

Yahoo奇摩 (國際通)  
2022-01-13 15:01:21



新冠肺炎疫情肆虐全球，連帶影響各國經濟發展。美國近日公布2021年1月至12月的消費者物價指數（CPI），上漲幅度達7%，飆升速度創下近40年新高，是1982年6月以來漲幅最大的一次。目前美國CPI已連續19個月上漲，近3個月CPI增幅則都高於6%。專家表示，疫情危機與工資壓力造成住房、汽車和能源價格上漲，是推動美國通貨膨脹走高的主因。市場預測美聯儲最早將於3月開始加息，以抑制消費、減緩通貨膨脹。

回顧1970 - 1980年代，當時美國全球競爭力開始降低，產生貿易逆差，美國又於1970年廢除金本位，造成美元大貶，種下通膨成型的因子。隨後由於1973、1979年陸續爆發2次石油危機，加上美國工會要求按通膨調整勞工薪資，使得原物料價格失控，薪資與物價也陷入螺旋式上升，使通膨變得嚴峻且難以解決。



# 美國物價新聞

## 美國生產力提升支撐長期經濟動能

科技進步生產力持續提升，將可支撐經濟成長及抑制長期通膨

美國總生產力變化(五年平均)



# Source

---

## 使用環境

- vscode

## 資料集來源

- <https://www.kaggle.com/datasets/elmoallistair/commodity-prices-196020>

## 使用套件

- 21
- Sklearn

# 資料集

## 物品名稱

年分

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Year	Cocoa	Coffee	Tea	Crude Oil	Coal	Natural Gas	Banana	Sugar	Orange	Barley	Maize	Sorghum	Rice	Wheat	Beef	Chicken	Lamb	Shrimps	Gold	Platinum	Silver	Cotton	R
2	1960	0.59	0.92	1.03	1.63		0.14	0.14	0.07	0.13	20.15	44.5	36.58	107.35	57.99	0.74	0.3		1.61	35.27	83.5	0.91	0.65	
3	1961	0.48	0.9	0.95	1.57		0.15	0.13	0.06	0.13	20.75	44.96	41.42	118.95	58.61	0.68	0.31		1.69	35.25	83.5	0.92	0.67	
4	1962	0.46	0.83	0.97	1.52		0.16	0.13	0.06	0.11	22.7	48.65	44.7	134.93	64.33	0.71	0.32		2.25	35.23	83.5	1.08	0.65	
5	1963	0.55	0.81	0.95	1.5		0.16	0.14	0.18	0.09	23.98	53.77	47.84	125.66	64.49	0.67	0.33		2	35.09	80.93	1.28	0.65	
6	1964	0.51	1.01	0.93	1.45		0.15	0.17	0.13	0.08	25.14	54.63	48.24	120.15	67.53	0.84	0.34		1.81	35.1	88.48	1.29	0.65	
7	1965	0.37	1	1	1.42		0.16	0.16	0.04	0.1	24.34	55.67	47.93	118.79	59.46	0.88	0.35		1.99	35.12	98.04	1.29	0.64	
8	1966	0.52	0.93	0.87	1.36		0.16	0.15	0.04	0.08	25.91	58.3	48.58	145.2	62.95	1.02	0.36		2.31	35.13	99.72	1.29	0.62	
9	1967	0.6	0.86	0.84	1.33		0.16	0.16	0.04	0.16	26.25	54.1	51.9	189.37	65.71	1.04	0.37		2.29	34.95	110.25	1.55	0.68	
10	1968	0.72	0.87	0.75	1.32		0.16	0.15	0.04	0.17	25.16	47.96	52.57	182.91	62.77	1.09	0.38		2.58	38.93	117.21	2.14	0.68	
11	1969	0.9	0.88	0.73	1.27		0.17	0.16	0.07	0.17	25.57	52.29	53.76	168.37	58.39	1.22	0.4		2.87	41.08	124.19	1.79	0.63	
12	1970	0.68	1.15	0.84	1.21	7.8	0.17	0.17	0.08	0.17	26.36	58.4	51.8	126.31	54.9	1.3	0.42		2.73	35.94	132.5	1.77	0.68	
13	1971	0.54	0.99	0.85	1.69	8.68	0.18	0.14	0.1	0.15	28.38	58.4	55.77	111.54	61.73	1.35	0.44	0.8	3.31	40.8	123.04	1.55	0.78	
14	1972	0.64	1.11	0.78	1.82	9.6	0.19	0.14	0.16	0.15	30.07	55.75	56.24	129.45	69.81	1.48	0.47	1.07	4.12	58.16	127.13	1.68	0.83	
15	1973	1.13	1.37	0.79	2.81	11.48	0.21	0.15	0.21	0.16	51.03	97.82	93.39	289.58	139.81	2.01	0.53	1.39	5	97.33	152.18	2.56	1.39	
16	1974	1.56	1.45	1.09	10.97	16.89	0.29	0.18	0.65	0.18	64.93	131.23	120.94	517.17	179.71	1.58	0.56	1.42	4.65	159.25	184.98	4.71	1.46	
17	1975	1.25	1.44	1.14	10.43	26.84	0.43	0.22	0.45	0.23	58.98	119.64	111.87	341.43	149.06	1.33	0.58	1.48	5.89	161.03	169.87	4.42	1.22	
18	1976	2.05	3.15	1.19	11.63	27.16	0.58	0.26	0.25	0.22	55.79	112.37	105.21	23	58	0.61	1.57	8.31	124.82	159.65	4.36	1.75		
19	1977	3.79	5.17	1.95	12.57	28.68	0.79	0.27	0.18	0.25	47.89	95.35	88.37	25	51	0.65	1.7	7.91	147.72	158.32	4.64	1.63		
20	1978	3.4	3.59	1.41	12.92	29.06	0.91	0.29	0.17	0.3	54.96	100.74	93.84	34	14	0.7	2.19	8.07	193.24	260.65	5.42	1.61		
21	1979	3.29	3.83	1.51	30.96	30.89	1.18	0.33	0.21	0.4	65.25	115.52	108.11	31	38	0.74	2.37	11.97	306.67	442.8	11.1	1.71		
22	1980	2.6	3.47	1.66	36.87	40.14	1.59	0.38	0.63	0.4	78.23	125.26	128.86	410.74	172.73	2.76	0.76	2.88	10.14	607.86	679.1	20.8	2.06	
23	1981	2.08	2.87	1.47	35.48	53.62	1.98	0.4	0.37	0.41	79.73	130.78	126.37	458.99	174.96	2.47	0.77	2.74	10.16	459.75	447.48	10.52	1.87	
24	1982	1.74	3.09	1.54	32.65	54.77	2.47	0.37	0.19	0.38	63.71	109.26	108.53	272.48	160.36	2.39	0.76	2.38	13.28	375.8	328.29	7.93	1.6	
25	1983	2.12	2.91	2.1	29.66	38.19	2.59	0.43	0.19	0.37	70.89	136.02	128.76	256.8	157.42	2.44	0.77	1.93	13.21	422.53	423.31	11.43	1.85	
26	1984	2.4	3.19	2.74	28.56	30.96	2.66	0.37	0.11	0.35	78.72	135.94	118.19	232.5	152.33	2.27	0.87	1.92	11.74	360.48	357.81	8.14	1.78	
commodity_prices																								

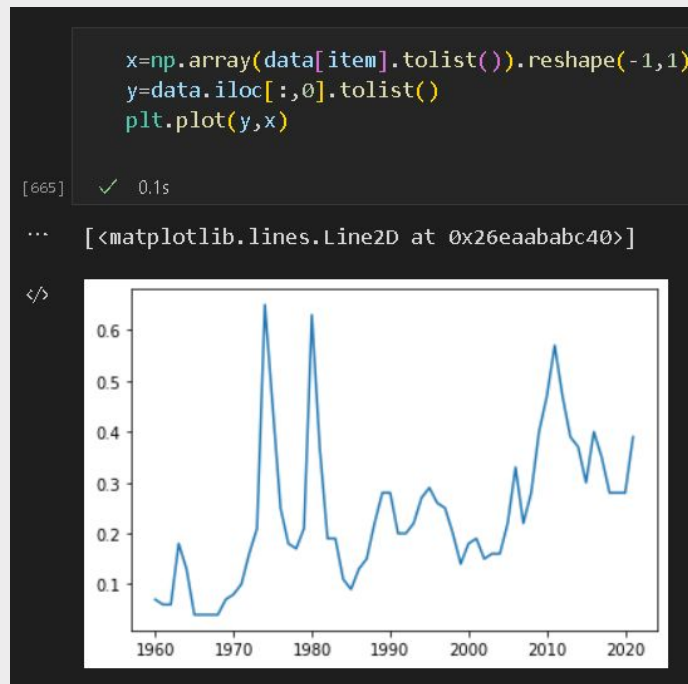
價格

# 讀取資料

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn import datasets

data = pd.read_csv("commodity_prices.csv")
#print(data)
item="Sugar" 把物品設為"糖"
```

[664] ✓ 0.4s



X軸:年分

Y軸:價格(美元)

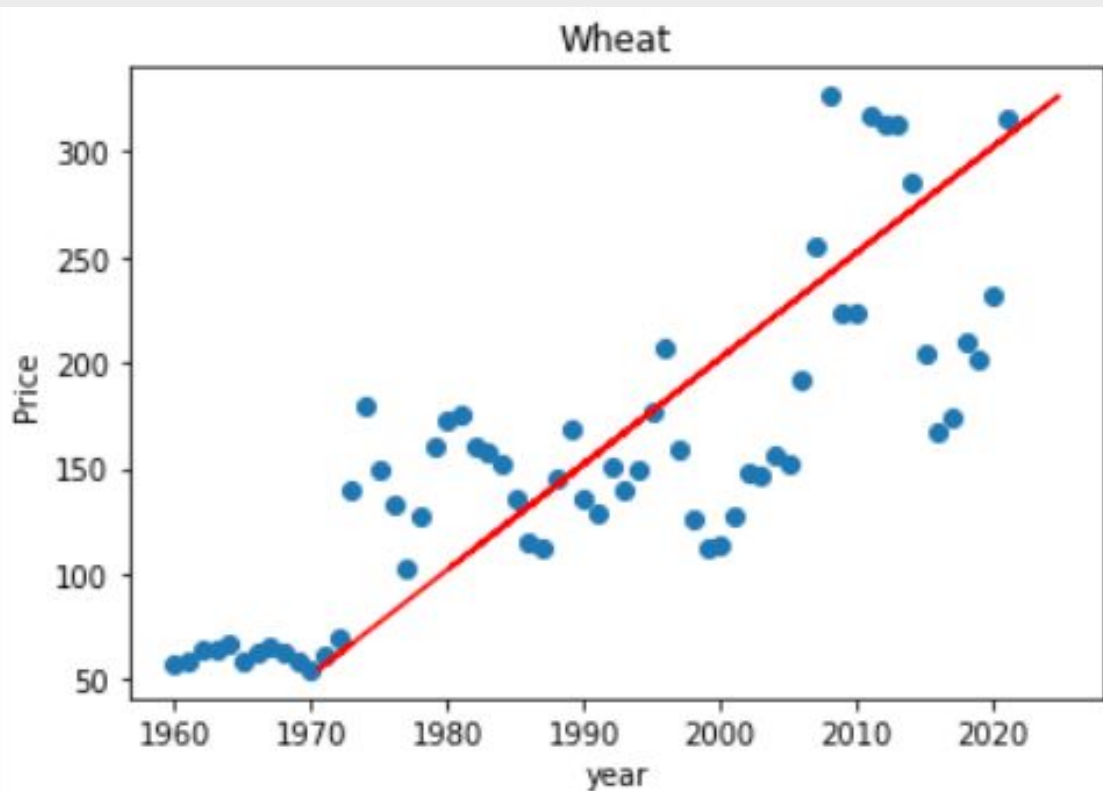
# 做出線性回歸模型

物品選擇

"Wheat"(大麥)

```
model = LinearRegression()  
model.fit(x,y)  
predict = model.predict(x)  
  
plt.title(item) # 圖的標題  
plt.xlabel("year") # x軸的名稱  
plt.ylabel("Price") # y軸的名稱  
plt.plot(predict,x,c="red")  
plt.scatter(y,x)  
plt.show()  
#plt.plot(y,x)|  
#plt.show()
```

[27] ✓ 0.2s

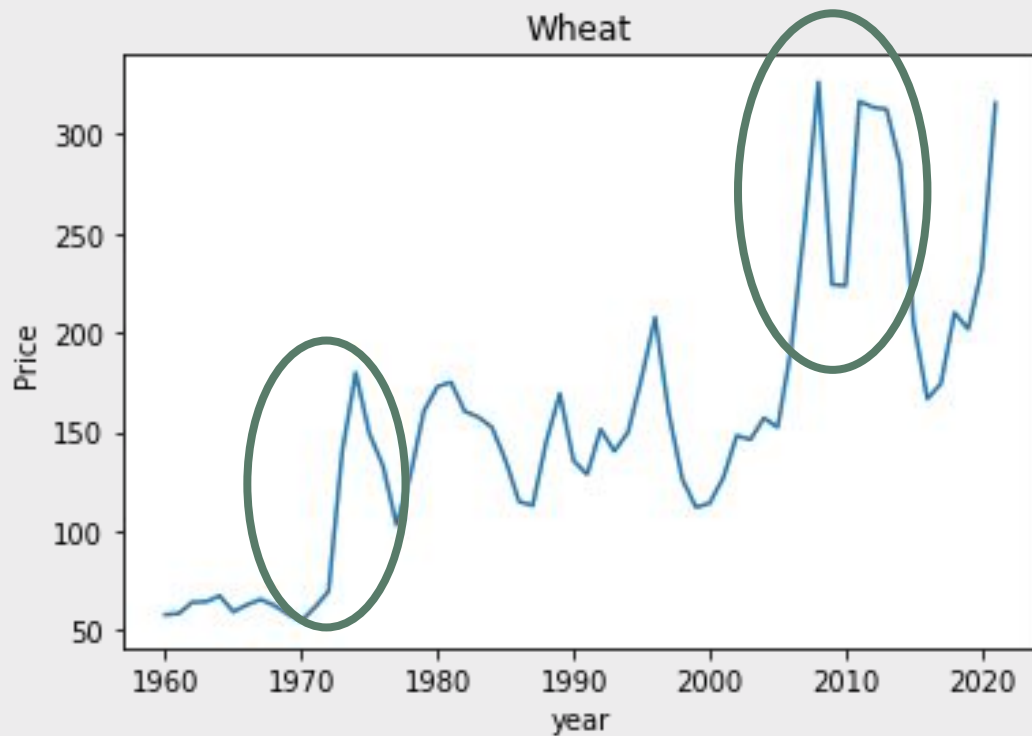




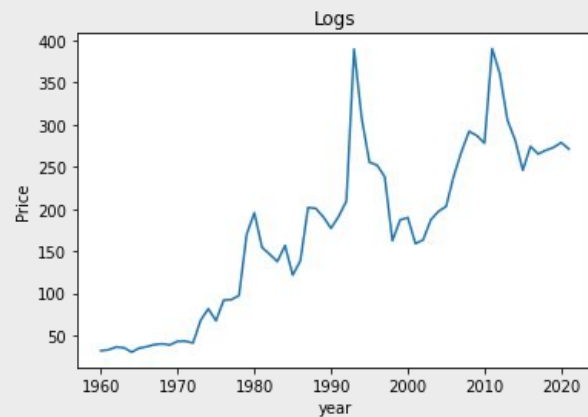
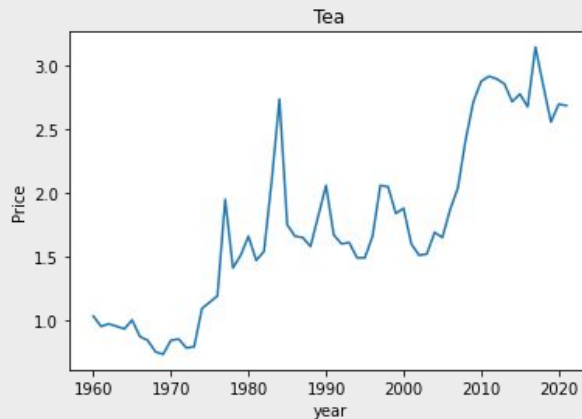
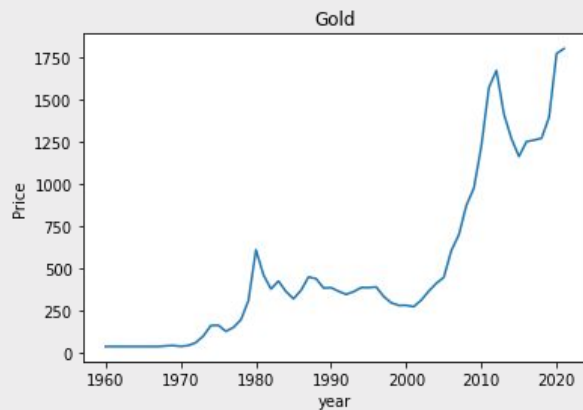
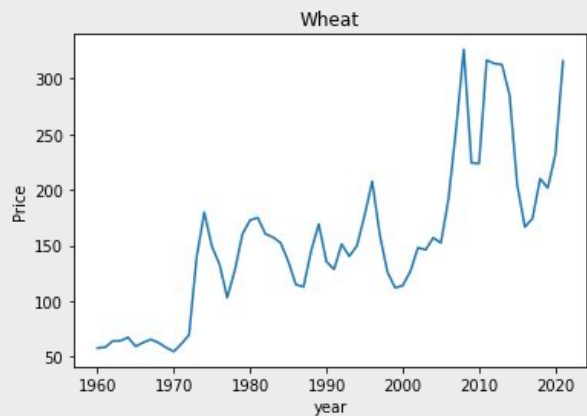
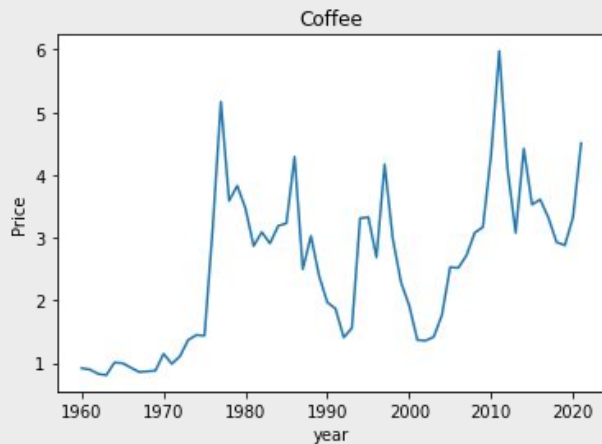
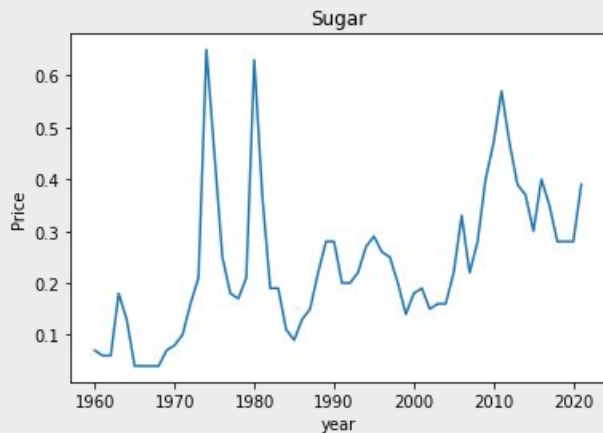
A close-up photograph of a person's hands holding a yellow orange. The orange has handwritten numbers in black marker: '13' at the top, '14' on the left, and 'x2' on the right. The background is a physics textbook with various diagrams and equations. A hand holding a white pen is visible on the right side of the frame. The text '但...數值夠準確嗎?' is overlaid in the center of the image.

但...數值夠準確嗎?

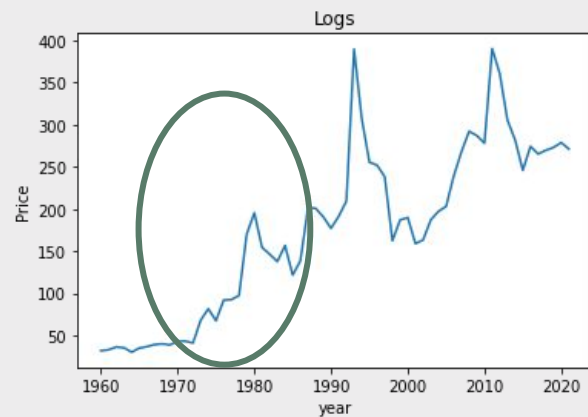
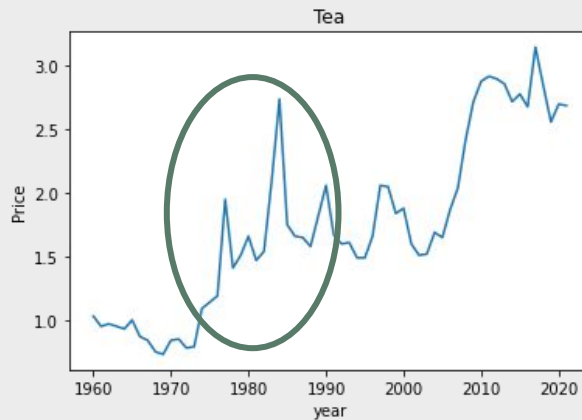
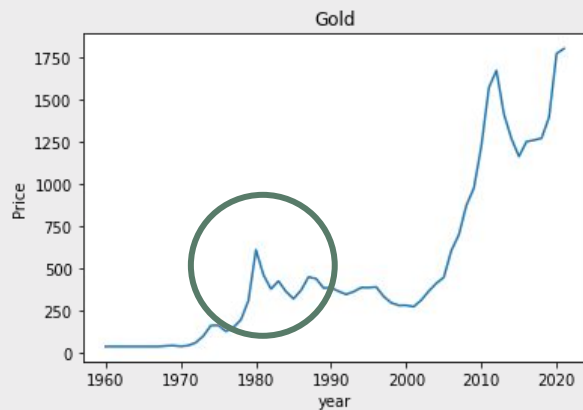
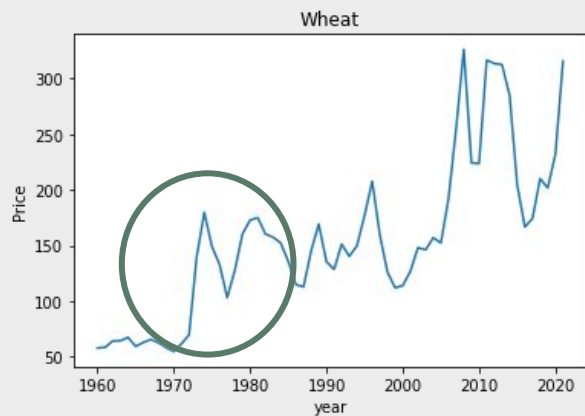
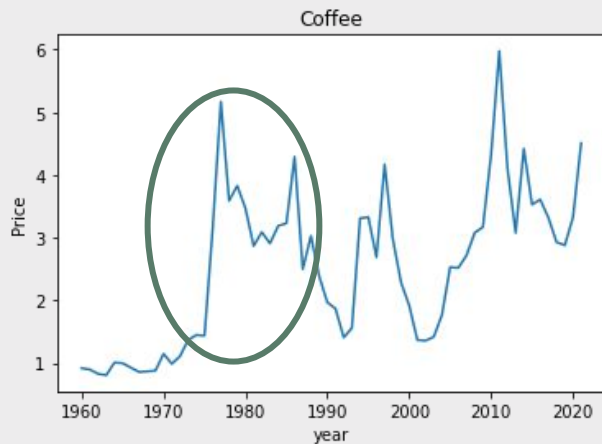
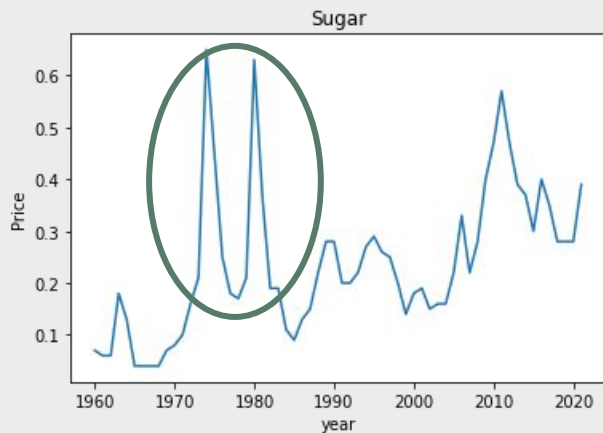
# 觀察價格數據



# 觀察各種物品價格變化



# 觀察各種物品價格變化



# 美國歷史通膨事件





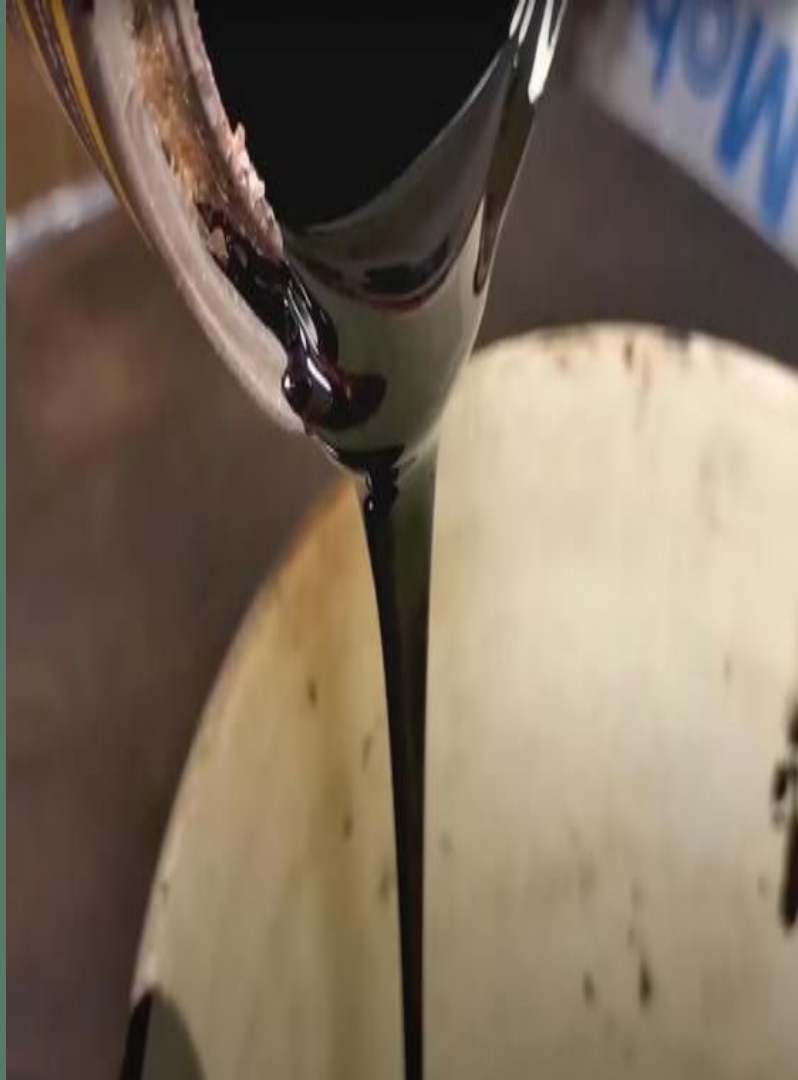
# 美國於1971年廢除金本位

金本位：每單位的貨幣價值  
等同於若干含重量的黃金

# 1973與1979石油危機

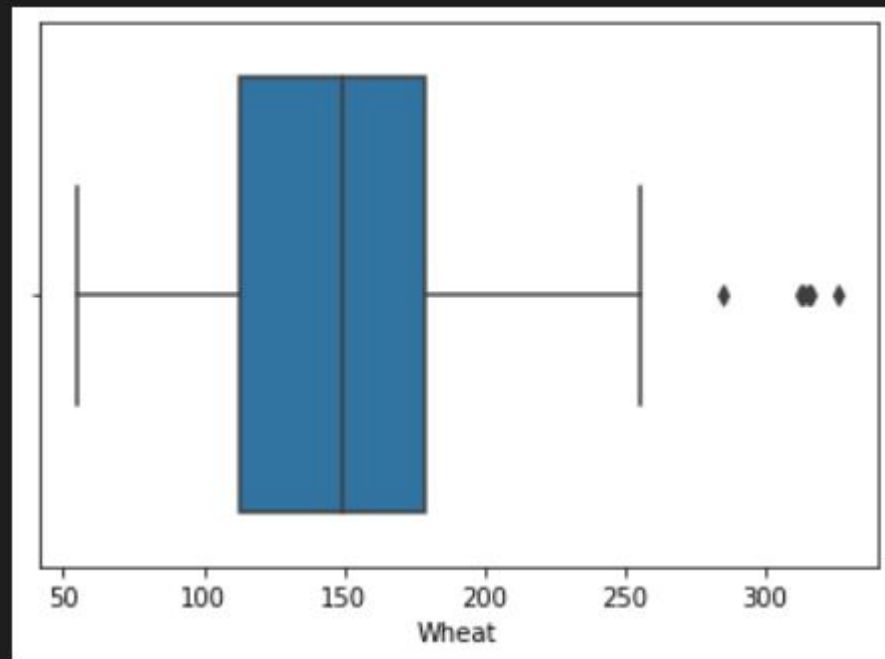
1973年贖罪日戰爭爆發  
(第四次以阿戰爭)

1979年伊朗革命爆發

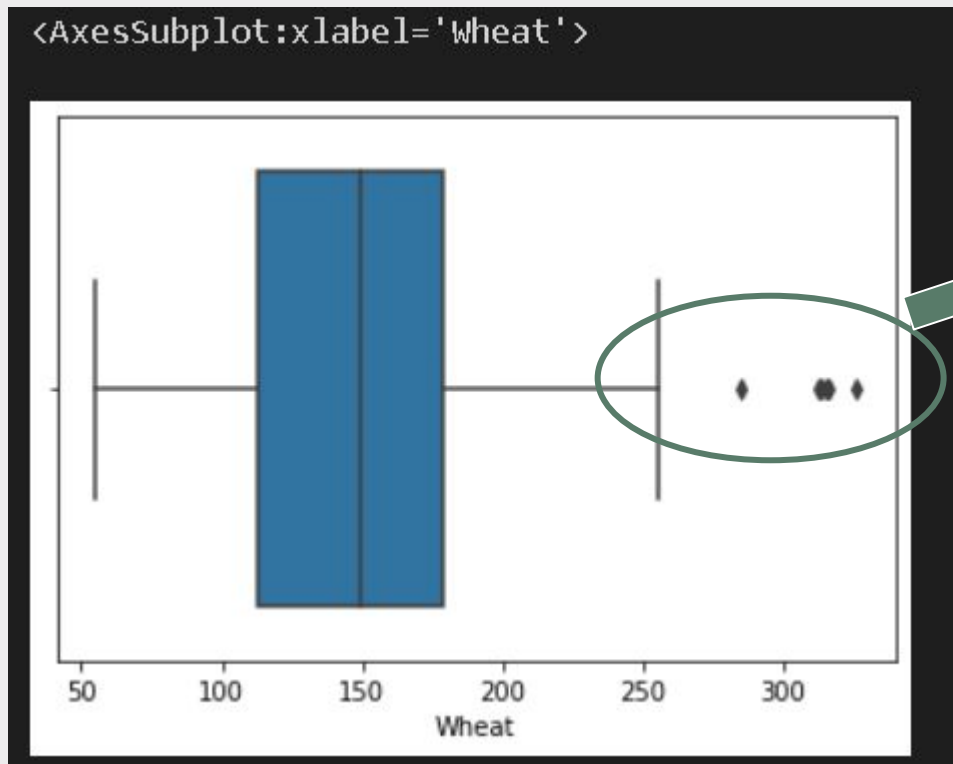


# 畫出箱型圖

```
<AxesSubplot:xlabel='wheat'>
```



# 畫出箱型圖



Outlier(異常值)

# 對異常值進行處理

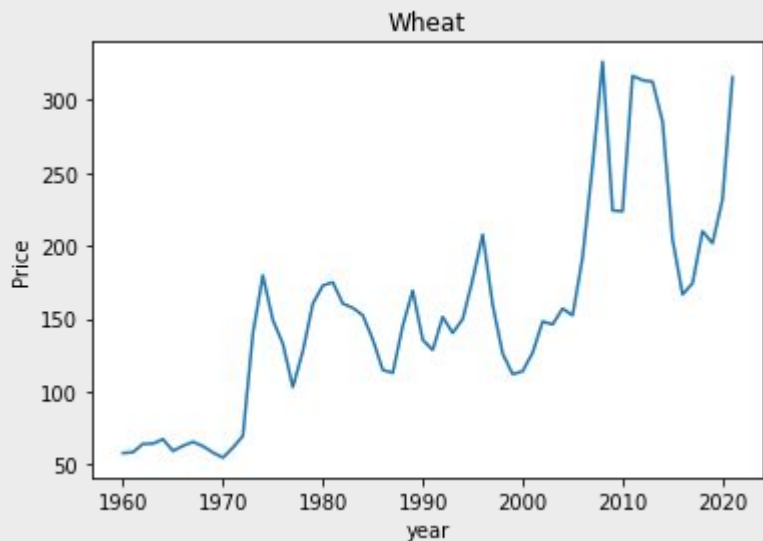
```
IQR = data[item].quantile(0.75) - data[item].quantile(0.25)

Lower_quantile = data[item].quantile(0.25) - (IQR * 1.5)
Upper_quantile = data[item].quantile(0.75) + (IQR * 1.5)
#print(Upper_quantile)
data_no_outlier=[]
year_no_outlier=[]
for i in range(0,len(data[item])):
    if(data[item][i]>Lower_quantile)&(data[item][i]<Upper_quantile):
        data_no_outlier.append(data[item][i])
        year_no_outlier.append(data["Year"][i])
newx=np.array(data_no_outlier).reshape(-1,1)
newy=np.array(year_no_outlier).reshape(-1,1)
```

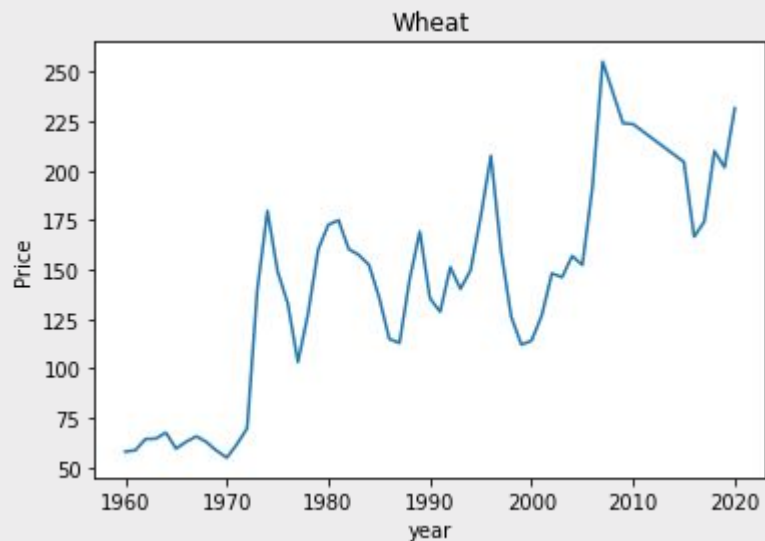


# 對異常值進行處理

Before



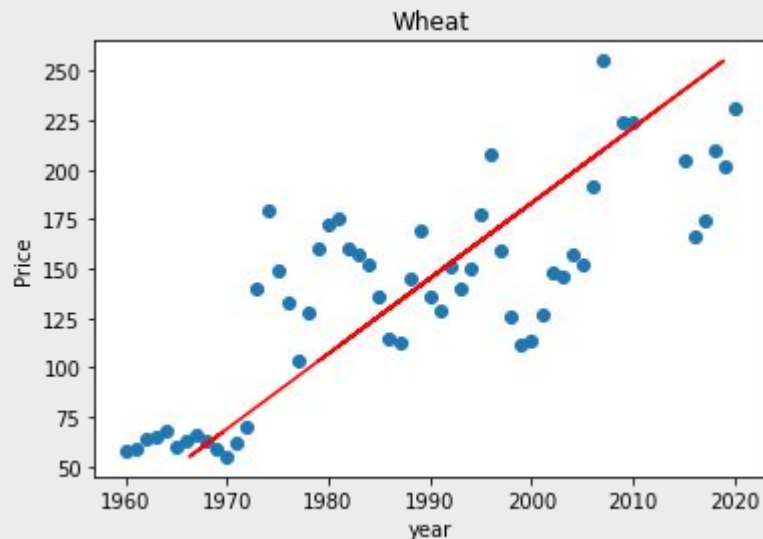
After



# 用去除異常值後的數據訓練模型

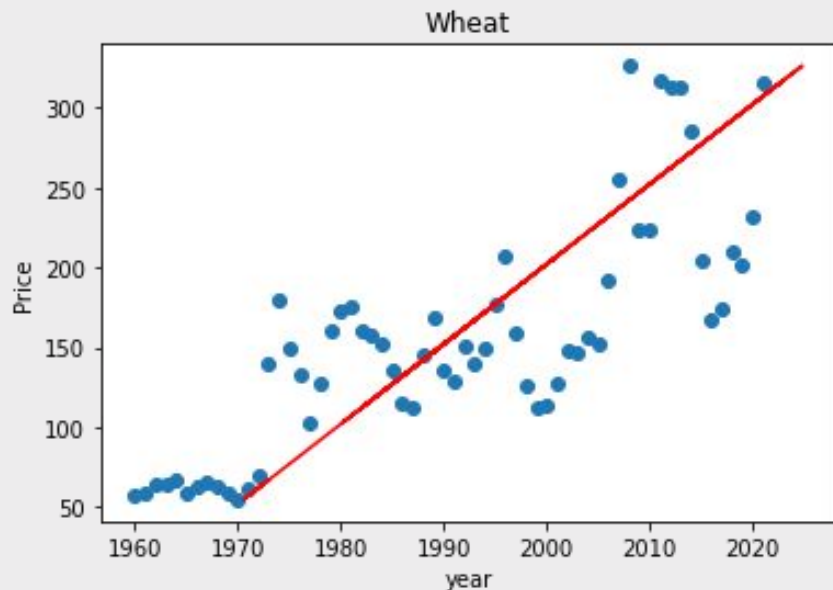
```
model2 = LinearRegression()  
model2.fit(newx,newy)  
predict = model2.predict(newx)  
plt.title(item) # 圖的標題  
plt.xlabel("year") # x軸的名稱  
plt.ylabel("Price") # y軸的名稱  
plt.plot(predict,newx,c="red")  
plt.scatter(newy,newx)  
plt.show()
```

✓ 0.1s

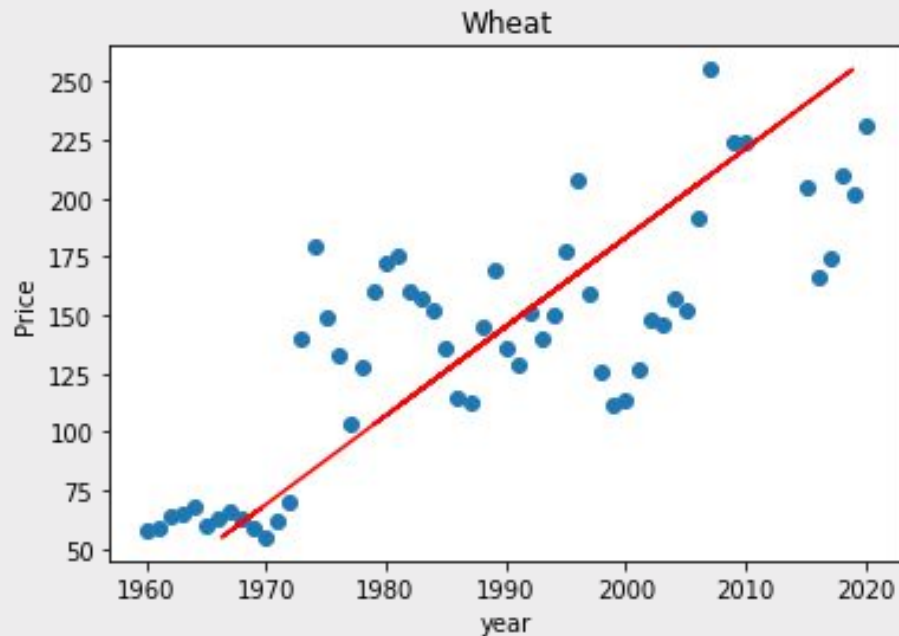


# 用去除異常值後的數據訓練模型

data\_with\_outlier



data\_without\_outlier



# 分數

```
from sklearn.model_selection import train_test_split
size=0.5
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = size, random_state = 0)
regressor = LinearRegression()
regressor.fit(x_train, y_train)
score1 = regressor.score(x_test, y_test)
print("data_with_outlier:",score1)
x_train2, x_test2, y_train2, y_test2 = train_test_split(newx, newy, test_size = size, random_state = 0)
regressor2 = LinearRegression()
regressor2.fit(x_train2, y_train2)
score2 = regressor2.score(x_test2, y_test2)
print("data_without_outlier:",score2)
```

[104] ✓ 0.4s

```
... data_with_outlier: 0.4995031618786253
data_without_outlier: 0.6114492775132305
```

# 分數

```
from sklearn.model_selection import train_test_split
size=0.5
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = size, random_state = 0)
regressor = LinearRegression()
regressor.fit(x_train, y_train)
score1 = regressor.score(x_test, y_test)
print("data_with_outlier:",score1)
x_train2, x_test2, y_train2, y_test2 = train_test_split(newx, newy, test_size = size, random_state = 0)
regressor2 = LinearRegression()
regressor2.fit(x_train2, y_train2)
score2 = regressor2.score(x_test2, y_test2)
print("data_without_outlier:",score2)
```

[104] ✓ 0.4s

... data\_with\_outlier: 0.4995031618786253

data\_without\_outlier: 0.6114492775132305

去除outlier後訓練的分數比原來高



# Google查2022年"Wheat"真實物價

## Commodity Prices

Data Card Code (0) Discussion (0)



36

New Notebook

- Rice (\$/kg)


Thailand, 5% broken, white rice (WR), milled, indicative price based on weekly surveys of export transactions, government standard, f.o.b. Bangkok

- Sorghum (\$/mt)

US, no. 2 milo yellow, Texas export bids for grain delivered to export elevators, rail-truck, f.o.b. Gulf ports

- Wheat (\$/mt)

US, no. 2 hard red winter Gulf export price; June 2020 backward, no. 1, hard red winter, ordinary protein, export price delivered at the US Gulf port for present or 30 days shipment



先去資料集提供的網站確認大麥單位為"mt"(metric ton)

# Google查2022年”Wheat”真實物價

2022 wheat mt price



**theGlobalEconomy.com**

Business and economic data for 200 countries

[Home](#)

[Countries](#)

[Indicators](#)

[Latest](#)

[Compare countries](#)

[Country rankings](#)

[Commodities](#)

[Outlook](#)

[Download data](#)

## Wheat price

最新價格為316

### Wheat price

- ▶ Banana price, Europe
- ▶ Banana price, US
- ▶ Orange prices
- ▶ Beef prices
- ▶ Chicken meat prices
- ▶ Sheep meat prices
- ▶ Mexican shrimp prices
- ▶ Sugar prices, EU

Commodity

Wheat price

Latest value

316.69

Reference

July 2022

Measure

USD per metric ton

Source

World Bank

# 2022年"Wheat"物價預測

用剛剛的模型對2022年大麥價格  
進行預測：

```
▶ ▾  
#截距  
a2=model2.intercept_  
#迴歸係數  
b2=model2.coef_  
answer2=((2022-a2)/b2)  
print(answer2)  
[135] ✓ 0.6s  
... [[266.96187585]]
```

# 我們發現的其他問題

# 2022年"Wheat"物價預測

用第一個模型對2022大麥價格  
進行預測(沒有去除outlier):

```
a1=model.intercept_  
b1=model.coef_  
answer=((2022-a1)/b1)  
print(answer)  
  
[134]  
... [311.91335905]
```



# 2022年"Wheat"物價預測

```
a1=model.intercept_  
b1=model.coef_  
answer=((2022-a1)/b1)  
print(answer)
```

[134]

... [311.91335905]

with\_outlier



```
#截距  
a2=model2.intercept_  
#迴歸係數  
b2=model2.coef_  
answer2=((2022-a2)/b2)  
print(answer2)
```

[135]

✓ 0.6s

... [[266.96187585]]

answer

Commodity	Wheat price
Latest value	316.69

without\_outlier

# 2022年"Wheat"物價預測

```
a1=model.intercept_  
b1=model.coef_  
answer=((2022-a1)/b1)  
print(answer)
```

[134]

... [311.91335905]

with\_outlier



answer

Commodity	Wheat price
Latest value	316.69

```
#截距  
a2=model2.intercept_  
#迴歸係數  
b2=model2.coef_  
answer2=((2022-a2)/b2)  
print(answer2)
```

[135]

✓ 0.6s

... [[266.96187585]]

without\_outlier

# 可能的原因

# 可能的原因：

1

近幾年經濟受疫情影響

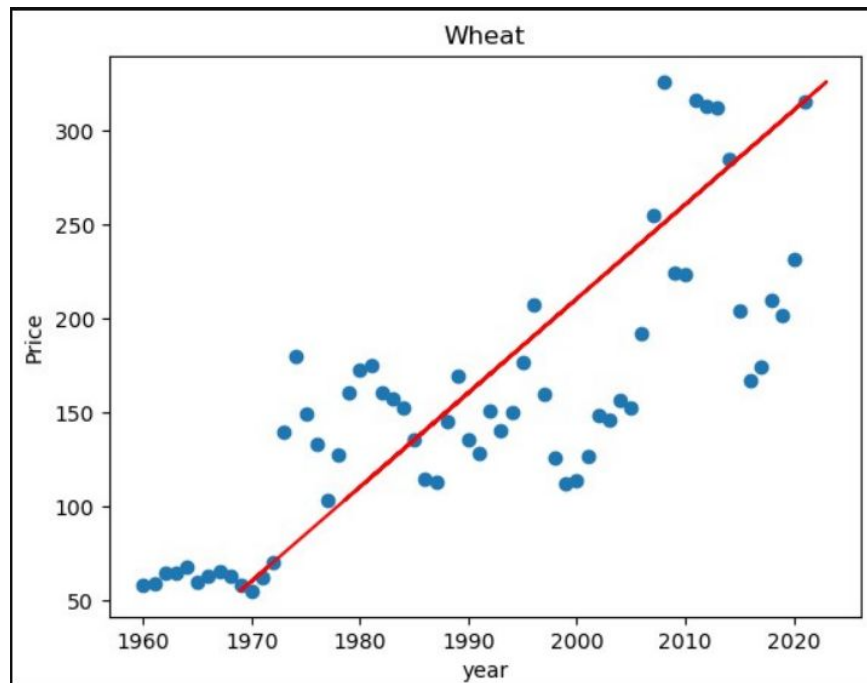
2

LinearRegression預測不夠準確

3

數據跟時事產生的落差

# 使用的其他模型: SVM



發現數據分布不會太分散 所以選擇不去除離群值

score: 0.6323646642703028  
2022: [[320.78112334]]

# 分數 & 預測結果

```
x=np.array(data[item].tolist()).reshape(-1,1)
y=data.iloc[:,0].tolist()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# 建立 kernel='linear' 模型
linearModel=svm.SVR(C=1, kernel='linear')
# 使用訓練資料訓練模型
linearModel.fit(x_train,y_train)
# 使用訓練資料預測分類
predict=linearModel.predict(x)

score = linearModel.score(x_test,y_test)
print("score:", score)

a1=linearModel.intercept_
b1=linearModel.coef_
answer=((2022-a1)/b1)
print('2022:',answer)
```

```
score: 0.6323646642703028
2022: [[320.78112334]]
```



**Thanks for listening!**