# 背景

A公司為一布料供應商，由於近年棉花供應短缺，以及其餘各項原料價格上漲，因此該公司決議重新制定各項產品的生產計劃。A公司使用過去五年內的數據以預測接下來一整年各個季度的市場需求，並以目標為最大化利潤的方式進行各季度各項產品的生產量決策。

# 預測

根據資料分布，觀察規律、趨勢等特徵後決定最適合的預測方式並實施之，對於找不出規律的數據，我們決定使用所有課堂上學過的forecasting方法處理，包括Moving average, Exponential Smoothing，Seasonal Exponential Smoothing，Holt-Winter，Regression，再根據一些常見的performance metrics，如MAE、MSE與MAPE或是R^2、F-value，來衡量哪一種方法最適合預測這份數據並進行預測，後續會一一說明各產品的預測概況。
在上述的前四種方法中，Moving average取前3期的平均，Exponential Smooting的alpha值定為0.4，Seasonal Exponential Smooting的alpha，beta值也都固定為0.4，而Holt-Winter則是alpha = 0.3，beta = 0.4，gamma = 0.3，最後要預測的額外期數為4期，代表一年的四個季度，同理，季節性因子(season length)也設為4。
額外說明：
使用較為複雜的Multiple Regression時，我們將CPI跟GDP也納入考量，以下為GDP，CPI的計算方式。

GDP：從The World Bank網站取得以現價美元為單位的2011年-2016年全球GDP，我們取2011年作為Base year，由於查找不到以季為單位的GDP，故我們根據The World Bank統計，算出各季GDP分別占每年總GDP的約 22%, 25%, 24%, 29%，依此比例算出2012-2016年的各季GDP，最後再分別除以2011年的GDP算出比例，以避免數值過大，以下為最終得出的數據：[0.225, 0.256, 0.245, 0.296, 0.231, 0.263, 0.252, 0.305, 0.237, 0.27, 0.259, 0.313, 0.224, 0.255, 0.244, 0.295, 0.228, 0.259, 0.248, 0.3]
參考資料：
https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?start=2000&fbclid=IwAR3ndmn6IHK3kbYfYjEv0c4eCadIfO3zTqk-Y45ysYSS7vBetyDn_rII6p4

CPI：從美國勞工局獲得2011-2016各季的CPI，同樣以2011年作為Base year，將2012-2016各季CPI除以2011年的CPI後得出比例，以下為最終得出的數據：

```
[1.028, 1.018, 1.016, 1.017, 1.042, 1.03, 1.029, 1.027, 1.055,
1.049, 1.045, 1.038, 1.052, 1.047, 1.044, 1.04, 1.06, 1.055,
1.052, 1.055]
```

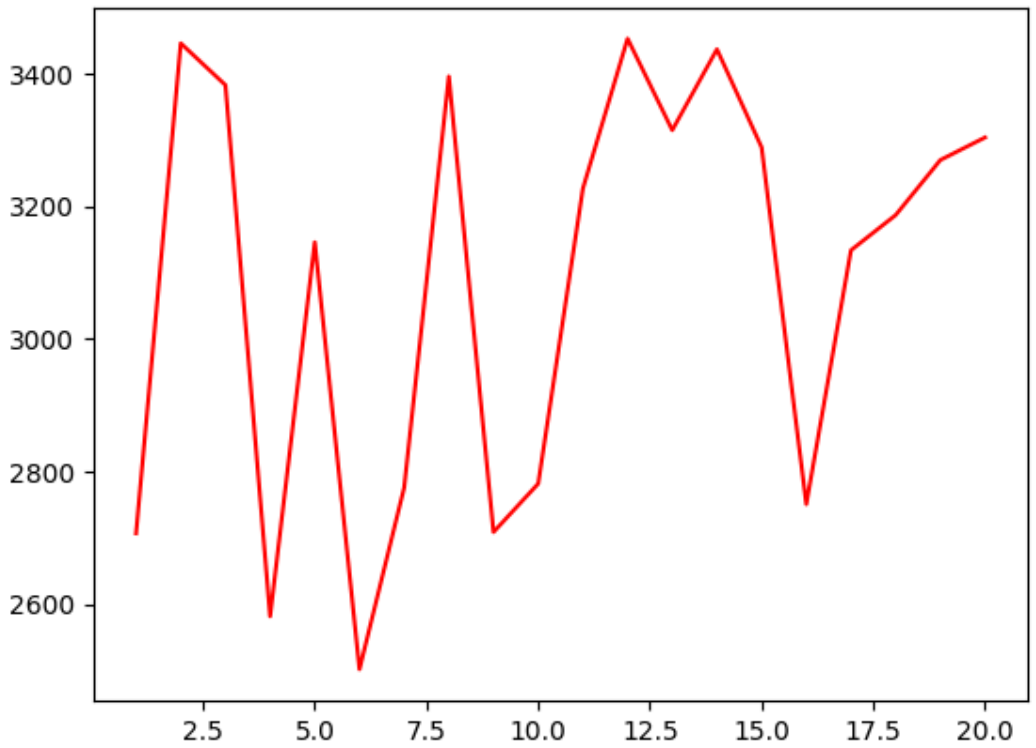
資料預先處裡:

我們在Kaggle上找到一份產品需求的資料，裡面記錄著大量產品的編號(Product_Code)、儲存倉庫(Warehouse)、類別(Category)、訂貨日期(Date)、需求量(Order_Demand)這五樣資訊，但由於這份資料集包含上百萬筆資料，所以我們決定採取隨機擷取其中7個類別，並在每個類別中的產品中隨機挑選5樣，共35個產品的策略，並進一步將所有產品的資料依照日期將每筆資料分類為第1~20期(Period)，每一期涵蓋一季度，以2012年第一季為第一期，2016年第四季為最後一期，不在此期間內的資料將被刪除，最後在將一個產品一期內的所有需求量加總，得到每個產品每一期的需求量，其中ProductID是我們幫個產品按Category的順序訂定，而Category的代號也由數字轉換成第一個到第七個的A~G。

資料來源:
https://www.kaggle.com/datasets/felixzhao/productdemandforecasting?select=Historical+Product+Demand.csv

**Product 1**

- 需求量分布:



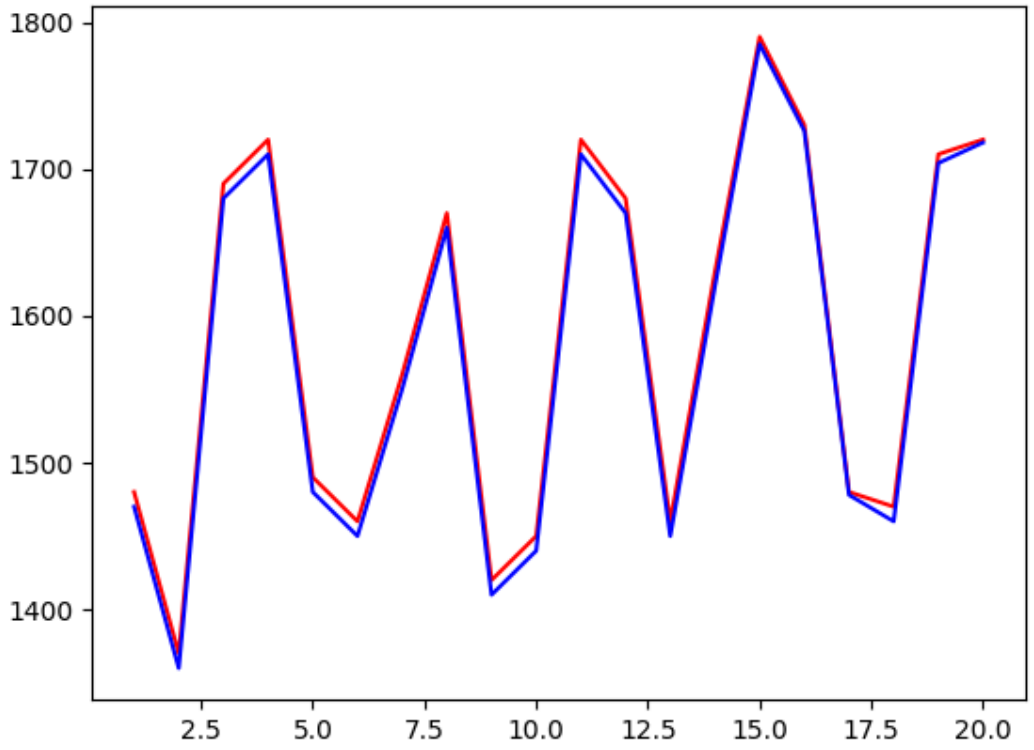
- 實際資料:
  Demand = [2707, 3446, 3383, 2582, 3146, 2502, 2776, 3396, 2709, 2782, 3227, 3453, 3315, 3437, 3289, 2751, 3134, 3187, 3270, 3304]
  Period = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
  Quarter = [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
  cost = 600*cpi_factor, price = 720*cpi_factor

- 預測方式:由於較難觀察出規律，因此分別使用Moving average，Exponential Smooting，Seasonal ES，Holt-Winter，Regression預測。

- 預測結果:
  L1將cpi剔除，最終MAPE為0.08，MAE為246.09，MSE為83264.34，三個指標中都是Lasso的表現最好，0.08的MAPE顯示這是一個精準度良好的模型。表現最差的模型是Holt-Winter model，MAPE為0.14，顯示季節性並不明顯。

**Product 2**

- 需求量分布:



- 實際資料:
  Demand_J = [1480, 1370, 1690, 1720, 1490, 1460, 1560, 1670, 1420, 1450, 1720, 1680, 1460, 1630, 1790, 1730, 1480, 1470, 1710, 1720]
  Demand_S = [1470, 1360, 1680, 1710, 1480, 1450, 1550, 1660, 1410, 1440, 1710, 1670, 1450, 1620, 1785, 1726, 1478, 1460, 1704, 1718]

- 預測方式:觀察出需求分布具有季節性，故使用Holt-Winter Model預測。

- 預測結果:
  WareHouse J的MAPE = 0.05，WareHouse S的MAPE = 0.05，兩者的MAE跟MSE亦十分相近，總體來說這是一個精度良好的模型，同時說明此商品的需求量與季節有密切的關聯。

各項參數與限制

由於預測結果為產品個數並非百分比，因此優先採用MAE、MSE作為預測模型準確度標準，且採用Multiple Regression的結果中$R^2$皆較小，故以MAPE作為次要標準，得出各產品各季度預測需求。

由於棉花供應短缺，故A公司一季度僅可從上游廠商購得25000公斤。
各類型商品產線皆有其產能限制，且除G類產品外，各項皆有單一季度生產下限1000個。
A公司共有四座倉庫，其倉儲上限皆不同。

綜合以上限制及商品成本、售價、各產品生產所需棉花量等既有數據，整理如下表:

表1

| ProductID | Warehouse | Cost | Price | Cutton(g/product) | Cutton(kg/season) | Productivity | Lower_Limit | 2017Q1 | 2017Q2 | 2017Q3 | 2017Q4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A | 600 | 720 | 66 | | | 1000 | 3254 | 3276 | 3278 | 3269 |
| 2 | J | 553 | 664 | 75 | | | 1000 | 1493 | 1477 | 1697 | 1697 |
| 2 | S | 553 | 664 | 75 | | | | 1489 | 1472 | 1695 | 1696 |
| 3 | J | 620 | 744 | 73 | | 24000 | 1000 | 3317 | 3352 | 3436 | 3512 |
| 4 | S | 623 | 697 | 66 | | | 1000 | 2219 | 2266 | 2289 | 2315 |
| 5 | S | 619 | 843 | 69 | | | 1000 | 1830 | 1816 | 2102 | 1934 |
| 5 | A | 619 | 843 | 69 | | | | 2742 | 2715 | 3146 | 2884 |
| 6 | S | 480 | 576 | 55 | | | 1000 | 6248 | 6323 | 6479 | 6299 |
| 7 | J | 464 | 557 | 48 | | | 1000 | 7090 | 7332 | 7512 | 7698 |
| 8 | S | 443 | 532 | 53 | | 80000 | 1000 | 8408 | 9101 | 8833 | 8292 |
| 9 | A | 450 | 540 | 47 | | | 1000 | 6358 | 6179 | 6070 | 6202 |
| 10 | J | 400 | 548 | 54 | | | 1000 | 11719 | 12246 | 12070 | 12012 |
| 11 | C | 215 | 258 | 24 | | | 1000 | 28713 | 21922 | 22306 | 31970 |
| 12 | C | 224 | 253 | 19 | | | 1000 | 10241 | 4770 | 7359 | 10537 |
| 13 | C | 211 | 253 | 16 | | 150000 | 1000 | 11729 | 28941 | 20210 | 24880 |
| 14 | C | 200 | 240 | 19 | | | 1000 | 19733 | 19811 | 19481 | 19675 |
| 15 | C | 179 | 215 | 24 | | | 1000 | 21833 | 21611 | 21348 | 21597 |
| 16 | A | 100 | 120 | 14 | | | 1000 | 29021 | 28802 | 28667 | 29311 |
| 17 | J | 140 | 190 | 6 | | | 1000 | 34767 | 35589 | 35519 | 35292 |
| 18 | S | 121 | 145 | 13 | | 162000 | 1000 | 22433 | 23144 | 22792 | 22790 |
| 19 | A | 72 | 86 | 14 | 25000 | | 1000 | 21633 | 21711 | 21659 | 21668 |
| 20 | J | 50 | 60 | 15 | | | 1000 | 26667 | 25689 | 25719 | 26025 |
| 21 | J | 189 | 225 | 19 | | | 1000 | 12880 | 11153 | 9262 | 7207 |
| 22 | J | 169 | 197 | 19 | | | 1000 | 29625 | 29625 | 29625 | 29625 |
| 23 | J | 150 | 185 | 22 | | 612000 | 1000 | 35277 | 40569 | 19139 | 8646 |
| 24 | J | 126 | 215 | 16 | | | 1000 | 459862 | 459862 | 459862 | 459862 |
| 25 | J | 123 | 145 | 18 | | | 1000 | 15333 | 35083 | 44708 | 31708 |
| 26 | A | 350 | 435 | 44 | | | 1000 | 14830 | 7873 | 12577 | 7918 |
| 26 | J | 350 | 435 | 44 | | | | 5727 | 5332 | 5342 | 5376 |
| 27 | J | 294 | 333 | 36 | | | 1000 | 25015 | 14581 | 17105 | 18251 |
| 28 | J | 250 | 360 | 41 | | 408000 | 1000 | 74752 | 109421 | 100284 | 109757 |
| 29 | J | 314 | 360 | 35 | | | 1000 | 24200 | 49400 | 69400 | 47667 |
| 30 | C | 324 | 477 | 39 | | | 1000 | 65710 | 65710 | 65710 | 65710 |
| 30 | J | 324 | 477 | 39 | | | | 77500 | 77500 | 77500 | 77500 |
| 31 | S | 765 | 1094 | 92 | | | 0 | 394 | 904 | 1255 | 851 |
| 31 | A | 765 | 1094 | 92 | | | 0 | 410 | 587 | 613 | 537 |
| 31 | J | 765 | 1094 | 92 | | | 0 | 278 | 278 | 278 | 278 |
| 32 | S | 800 | 960 | 90 | | 9000 | 0 | 169 | 169 | 169 | 169 |
| 33 | A | 783 | 940 | 93 | | | 0 | 1881 | 1881 | 1881 | 1881 |

表2

| Warehouse | Capicity |
|---|---|
| A | 110000 |
| C | 235000 |
| J | 1715000 |
| S | 60000 |

# 最佳化模型

我們使用scipy來幫助我們做決策，共有35項產品，並設定42個決策變數(包含儲存在不同倉庫的相同產品)，來決定在各種限制下該怎麼調整我們的生產策略。

## 限制條件：

- 原物料供應限制：由於所有產品皆須用到棉花，但棉花供應短缺，所以總產量中使用的棉花不能超過棉花供應量，限制式如下：

```
def constraint1(x):
    return 25000 * 1000 - sum(material[i] * x[i] for i in range(I))
```

- 每條產線的產能限制：由於生產設備、人力資源、時間成本等限制，每條產線的產能有上限，該條產線的總生產量不能超過產能上限，限制式如下：

```
def constraint2(x):
    return 24000 - (sum(x[0:7]))

def constraint3(x):
    return 80000 - (sum(x[7:12]))

def constraint4(x):
    return 150000 - (sum(x[12:17]))

def constraint5(x):
    return 162000 - (sum(x[17:22]))

def constraint6(x):
    return 612000 - (sum(x[22:27]))

def constraint7(x):
    return 408000 - (sum(x[27:34]))

def constraint8(x):
    return 9000 - (sum(x[34:42]))
```

- 每條產線的生產量下限：為了維持基本的生產量，同時避免模型出現負數結果，我們設定了生產量下限，限制式如下：

```
def constraint9(x):
    return x - 1000

def constraint10(x):
    return x[1] + x[2] - 1000

def constraint11(x):
    return x[5] + x[6] - 1000

def constraint12(x):
    return x[27] + x[28] - 1000

def constraint13(x):
    return x[32] + x[33] - 1000

def constraint14(x):
    return x
```

- 倉儲容量限制：A, C, J, S四個倉儲個別設置容量上限，限制式如下：

```
    def constraint15(x):
        return 110000 - (x[0]+x[6]+x[10]+x[17]+x[20]+x[27]+x[35]+x[38]+x[39])

    def constraint16(x):
        return 235000 - (x[12]+x[13]+x[14]+x[15]+x[16]+x[32])

    def constraint17(x):
        return 1715000 - (x[1]+x[3]+x[8]+x[11]+x[18]+x[21]+x[22]+x[23]+x[24]+x[25]+x[26]+x[28]+x[29]+x[30]+x[31]+x[33]+x[36]+x[41])

    def constraint18(x):
        return 60000 - (x[2]+x[4]+x[5]+x[7]+x[9]+x[19]+x[34]+x[37]+x[40])
```

## 目標函式：

我們的目標是找出利潤最大化的情況下，每項產品的生產量分配，而銷售量我們取實際生產量和預測需求量中較小的值，且當產品生產超過需求量時，會產生未出售的成本。收入減成本即為我們的利潤。

```
def objectiveQ1(x):
        # 計算總收入
        revenueQ1 = sum(price[i] * min(x[i], demandQ1[i]) for i in range(I))

        # 計算總成本
        total_costQ1 = sum(cost[i] * x[i] for i in range(I))+sum(price[i] * 0.1 * max(x[i]-demandQ1[i], 0) for i in range(I))

        # 目標函數為總收入減去總成本
        objQ1 = revenueQ1 - total_costQ1

        return -objQ1
```

# 結果

表3

| ProductID | Warehouse | SupplyQ1 | Predicted_ProfitQ1 | SupplyQ2 | Predicted_ProfitQ2 | SupplyQ3 | Predicted_ProfitQ3 | SupplyQ4 | Predicted_ProfitQ4 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | A | 3254 | | 3276 | | 3278 | | 3269 | |
| 2 | J | 1493 | | 1000 | | 1689 | | 1614 | |
| 2 | S | 1488 | | 1000 | | 1591 | | 1696 | |
| 3 | J | 3317 | | 1429 | | 3436 | | 3512 | |
| 4 | S | 1000 | | 1000 | | 1000 | | 1000 | |
| 5 | S | 1830 | | 1816 | | 2102 | | 1934 | |
| 5 | A | 2742 | | 2715 | | 3146 | | 2884 | |
| 6 | S | 6248 | | 6323 | | 6479 | | 6299 | |
| 7 | J | 7090 | | 7331 | | 7511 | | 7679 | |
| 8 | S | 8407 | | 3220 | | 7808 | | 1515 | |
| 9 | A | 6358 | | 6179 | | 6070 | | 6202 | |
| 10 | J | 11719 | | 12246 | | 12070 | | 12012 | |
| 11 | C | 28677 | | 11349 | | 14133 | | 31965 | |
| 12 | C | 10234 | | 1053 | | 1000 | | 1000 | |
| 13 | C | 11729 | | 28903 | | 20210 | | 24880 | |
| 14 | C | 19732 | | 19809 | | 19481 | | 19675 | |
| 15 | C | 12287 | | 3225 | | 1280 | | 4088 | |
| 16 | A | 1620 | | 1000 | | 11951 | | 1000 | |
| 17 | J | 34767 | | 35589 | | 35519 | | 35292 | |
| 18 | S | 7300 | | 10211 | | 13273 | | 20863 | |
| 19 | A | 2063 | 84421696 | 6981 | 86956493 | 1001 | 86805760 | 1153 | 86924071 |
| 20 | J | 1003 | | 1491 | | 1000 | | 3877 | |
| 21 | J | 12846 | | 11153 | | 9262 | | 7207 | |
| 22 | J | 9291 | | 9152 | | 1000 | | 1104 | |
| 23 | J | 35197 | | 26159 | | 14282 | | 1081 | |
| 24 | J | 459862 | | 459862 | | 459862 | | 459862 | |
| 25 | J | 2500 | | 10760 | | 10495 | | 6229 | |
| 26 | A | 14830 | | 7857 | | 12574 | | 7918 | |
| 26 | J | 5727 | | 5269 | | 5342 | | 5376 | |
| 27 | J | 10181 | | 1125 | | 1557 | | 1027 | |
| 28 | J | 74752 | | 109421 | | 100284 | | 109757 | |
| 29 | J | 11798 | | 18031 | | 19852 | | 22181 | |
| 30 | C | 65710 | | 65710 | | 65710 | | 65710 | |
| 30 | J | 77500 | | 77500 | | 77500 | | 77500 | |
| 31 | S | 1000 | | 1000 | | 1255 | | 1000 | |
| 31 | A | 1000 | | 1000 | | 1000 | | 1000 | |
| 31 | J | 1000 | | 1000 | | 1000 | | 1000 | |
| 32 | S | 1000 | | 1000 | | 1000 | | 1000 | |
| 33 | A | 1704 | | 1692 | | 1488 | | 1388 | |
| 34 | A | 1296 | | 1001 | | 1256 | | 1055 | |
| 35 | S | 1000 | | 1000 | | 1000 | | 1000 | |
| 35 | J | 1000 | | 1000 | | 1000 | | 1000 | |