Session 10: Simulation Modeling II (Solutions Only)

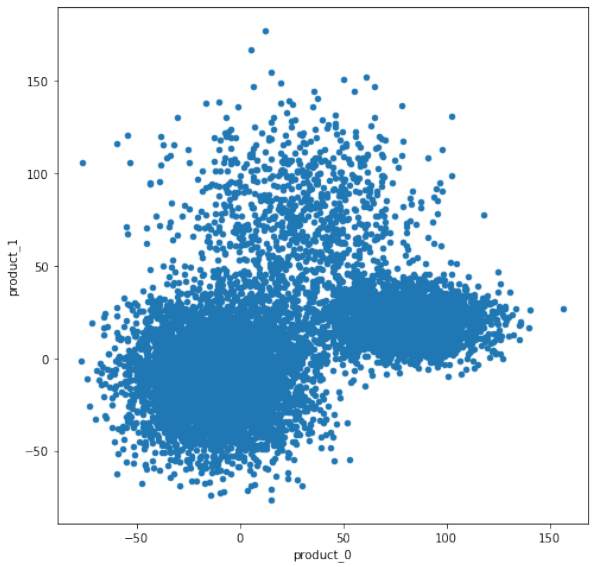
```
[2]: import pandas as pd
     from scipy.stats import norm
     import numpy as np
     data=[]
     dist0={'A':norm(30,30),'B':norm(80,20),'C':norm(-10,20)}
     dist1={'A':norm(70,30),'B':norm(20,10),'C':norm(-10,20)}
     for i in range(10000):
         segment=np.random.choice(['A','B','C'],p=[.1,.3,.6])
         data.append([segment,dist0[segment].rvs(),dist1[segment].rvs()])
     valuations=pd.DataFrame(data,columns=['segment','product_0','product_1'])
```

Once you have completed this part, you should be able to run the following code and obtain similar outputs.

```
[3]: valuations.head()
```

	segment	product_0	product_1
0	C	-1.139081	-20.163528
1	C	-9.411392	-10.416227
2	C	-22.082534	-23.561772
3	C	22.198945	-24.888582
4	C	-41.443531	-22.556809

```
[22]: valuations.plot(x='product_0',y='product_1',kind='scatter',figsize=(8,8))
```



Part III. Optimization

A. Brute Force Search of Optimal Price Vector

Search through all combinations of prices (p_0, p_1), where p_0 and p_1 are taken from from `range(0,200,5)`, which is equivalent to the list `[0,5,10,...,195]`. Identify the revene maximizing combination of prices and the estimated revenue.

```
[13]: bestP0,bestP1,bestRev=0,0,0
      for p0 in range(0,200,5):
          for p1 in range(0,200,5):
              d0,d1=demand2([p0,p1],valuations)
              revenue=p0*d0+p1*d1
              if revenue>bestRev:
                  bestP0,bestP1,bestRev=p0,p1,revenue
      print(f'Best combination is p0={bestP0}, p1={bestP1}, with revenue=${bestRev}.')
```

```
Best combination is p0=60, p1=60, with revenue=$194160.
```

B. Responding to Competitor Pricing

Suppose now that Model 0 is sold by a competitor, and only revenue from Model 1 counts. Find the optimal price to charge if the competitor prices Model 0 at $80. (Optional: plot a graph of what is the best price for Model 1 given any price for Model 0, as well as the resultant revenue.)

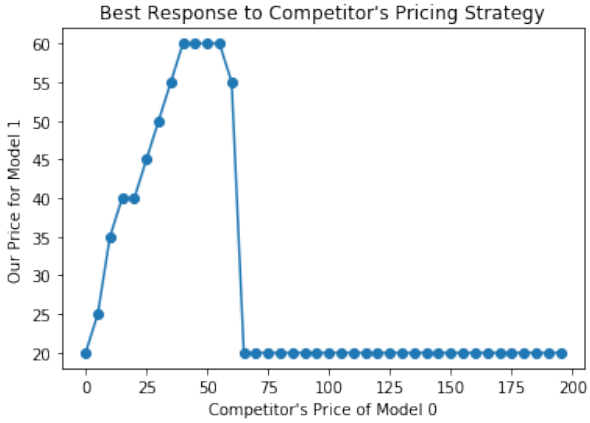
```
[14]: p0=80
      bestP1,bestRev=0,0
      for p1 in range(0,200,5):
          d0,d1=demand2([p0,p1],valuations)
          revenue=p1*d1
          if revenue>bestRev:
              bestP1,bestRev=p1,revenue
      print(f'Best response to p0={p0} is p1={bestP1}, with revenue=${bestRev}.')
```

```
Best response to p0=80 is p1=20, with revenue=$48100.
```

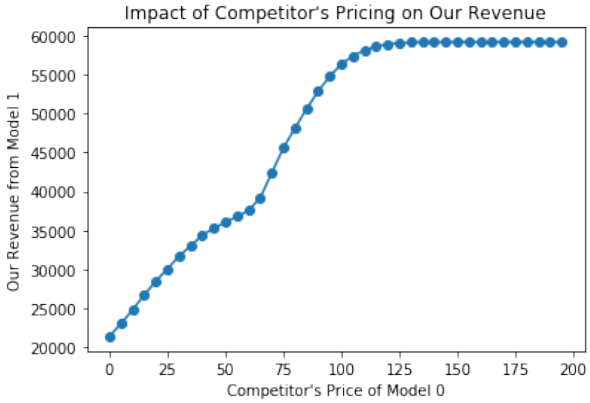
```
[15]: def bestResponse(p0):
          bestP1,bestRev=0,0
          for p1 in range(0,200,5):
              d0,d1=demand2([p0,p1],valuations)
              revenue=p1*d1
              if revenue>bestRev:
                  bestP1,bestRev=p1,revenue
          return bestP1,bestRev

      import matplotlib.pyplot as plt
      p0List=range(0,200,5)
      result=pd.DataFrame([bestResponse(p0) for p0 in p0List],columns=['bestP1','revenue'])
      result.head()
```

	bestP1	revenue
0	20	21420
1	25	23050
2	35	24815
3	40	26720
4	40	28480

[16]:
```
plt.plot(p0List,result['bestP1'],'o-')
plt.xlabel("Competitor's Price of Model 0")
plt.ylabel('Our Price for Model 1')
plt.title("Best Response to Competitor's Pricing Strategy")
plt.show()
```



[17]:
```
plt.plot(p0List,result['revenue'],'o-')
plt.xlabel("Competitor's Price of Model 0")
plt.ylabel('Our Revenue from Model 1')
plt.title("Impact of Competitor's Pricing on Our Revenue")
plt.show()
```



C. Benefit of Price Discrimination

Suppose that the company can observe which segment each customer belongs to, and charge separate prices to each segment. What would be the optimal prices for each segment and what would be the additional revenue from this flexibility?

Hint: Filter the "valuations" DataFrame by whether the segment is A, B or C, and plug in the filtered DataFrames to the previous analysis. The following code shows one way to filter DataFrames:

```
valuations.query('segment=="A"')
```

```
[18]: def optimizeTwoPrices(valuations):
          bestP0,bestP1,bestRev=0,0,0
          for p0 in range(0,200,5):
              for p1 in range(0,200,5):
                  d0,d1=demand2([p0,p1],valuations)
                  revenue=p0*d0+p1*d1
                  if revenue>bestRev:
                      bestP0,bestP1,bestRev=p0,p1,revenue
          return bestP0,bestP1,bestRev
      print('Segment A:')
      bestP0,bestP1,bestRevA=optimizeTwoPrices(valuations.query('segment=="A"'))
      print(f'Best combination is p0={bestP0}, p1={bestP1}, with revenue=${bestRevA}.')
```

```
Segment A:
Best combination is p0=45, p1=60, with revenue=$42945.
```

```
[19]: print('Segment B:')
      bestP0,bestP1,bestRevB=optimizeTwoPrices(valuations.query('segment=="B"'))
      print(f'Best combination is p0={bestP0}, p1={bestP1}, with revenue=${bestRevB}.')
```

```
Segment B:
Best combination is p0=60, p1=30, with revenue=$153540.
```

```
[20]: print('Segment C:')
      bestP0,bestP1,bestRevC=optimizeTwoPrices(valuations.query('segment=="C"'))
      print(f'Best combination is p0={bestP0}, p1={bestP1}, with revenue=${bestRevC}.')
```

```
Segment C:
Best combination is p0=15, p1=15, with revenue=$18180.
```

```
[21]: bestRev=optimizeTwoPrices(valuations)[-1]
      benefit=bestRevA+bestRevB+bestRevC-bestRev
      print(f'Potential benefit of price discrimination is ${benefit}.')
```

```
Potential benefit of price discrimination is $20505.
```