

決策科學 Homework3 (Group8)

資管碩一 109356011 簡琬玲

資管碩一 109356012 劉鎧瑄

資管碩一 109356023 商肯豪

資管碩一 109356024 蘇品維

資管碩一 109356040 李祈寬

Q1:

```
catch_fish=10000#模擬1000次出海捕魚
boat_capacity=3800#滿船容量
cost=7200#operation cost
corrgr=c(-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8)#correlation table

strategy_table<-data.frame(matrix(ncol = 4, nrow = 0))#設一個空的策略表，將來會存放s1~s4
x <- c("s1", "s2", "s3", "s4")#命名
colnames(strategy_table) <- x

#模擬Rick和morty去抓魚
R_catch=runif(catch_fish, min = 0.6*boat_capacity, max = 0.9*boat_capacity)
M_catch=rtri(catch_fish,0.5*boat_capacity,1*boat_capacity,0.75*boat_capacity)

#g和r市場的需求
dg=rtri(catch_fish,4000,8000,7000)
dr=rtri(catch_fish,4800,7200,6300)

library(MASS)

for (cor in corrgr){
  covxy=cor*0.35*0.25 #算出g市場和r市場價格的共變異數
  sigma <- matrix(c(0.35**2,covxy,covxy,0.25**2), 2, 2)#製作matrix
  mean <- c(3.5,3.65)
  p <- mvrnorm(catch_fish, mu = mean, Sigma = sigma)#模擬兩市場的價格
  price_g=p[,1]#擷取g市場的價格
  price_r=p[,2]#擷取r市場的價格
  for (i in c(1:10000)){
    #模擬strategy的獲利
    strategy1=min(R_catch[i],dg[i])*price_g[i]-7200+min(M_catch[i],dr[i])*price_r[i]-7200
    #R船員進入G市場,M船員進入R市場
    strategy2=min(R_catch[i],dr[i])*price_r[i]-7200+min(M_catch[i],dg[i])*price_g[i]-7200
    #R船員進入R市場,M船員進入G市場
    strategy3=min(R_catch[i]+M_catch[i],dg[i])*price_g[i]-7200-7200
    #R船員進入G市場,M船員進入G市場
    strategy4=min(R_catch[i]+M_catch[i],dr[i])*price_r[i]-7200-7200
    #R船員進入R市場,M船員進入R市場
    strategy=c(strategy1,strategy2,strategy3,strategy4)#加入空的table
    strategy_table=rbind(strategy_table, strategy)
  }

}

print("finish")

draw_table<-data.frame()#儲存要畫圖的資料
```

```

cor_index=c(0,10000,20000,30000,40000,50000,60000,70000,80000,90000)#因為1-10000是corr=-0.8的answer，10001~20000是
#corr=-0.6的answer，所以紀錄要取出的index範圍

#(1)
for (i in 1:4){#巡迴s1~s4
  print(i)
  s=strategy_table[i]
  for(j in 1:(length(cor_index)-1)){
    #取出各corr代表的資料再取平均
    m=sum(s[(cor_index[j]+1):cor_index[j+1],])/10000
    draw_table=rbind(draw_table, m)
  }
}

s1=draw_table[1:9,]
s2=draw_table[10:18,]
s3=draw_table[19:27,]
s4=draw_table[28:36,]

#第一層畫圖程式
corrgr=c(-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8)
library(ggplot2)
ggplot()+geom_line(aes(corrgr,s1),colour="blue")+geom_line(aes(corrgr,s2),colour="green")+geom_line(aes(corrgr,s3),colour="black")+geom_line(aes(corrgr,s4),colour="red")

#(2)
draw_table<-data.frame()
for (i in 1:4){
  print(i)
  s=strategy_table[i]
  for(j in 1:(length(cor_index)-1)){
    a=s[(cor_index[j]+1):cor_index[j+1],]
    q=quantile(a, probs =0.05)#取出q cvar5%
    a=a[a<=q]#取出小於q的所有值
    m=mean(a)#取平均
    draw_table=rbind(draw_table, m)
  }
}

s1=draw_table[1:9,]
s2=draw_table[10:18,]
s3=draw_table[19:27,]
s4=draw_table[28:36,]

ggplot()+geom_line(aes(corrgr,s1),colour="blue")+geom_line(aes(corrgr,s2),colour="green")+geom_line(aes(corrgr,s3),colour="black")+geom_line(aes(corrgr,s4),colour="red")

```

PS.以下圖示:

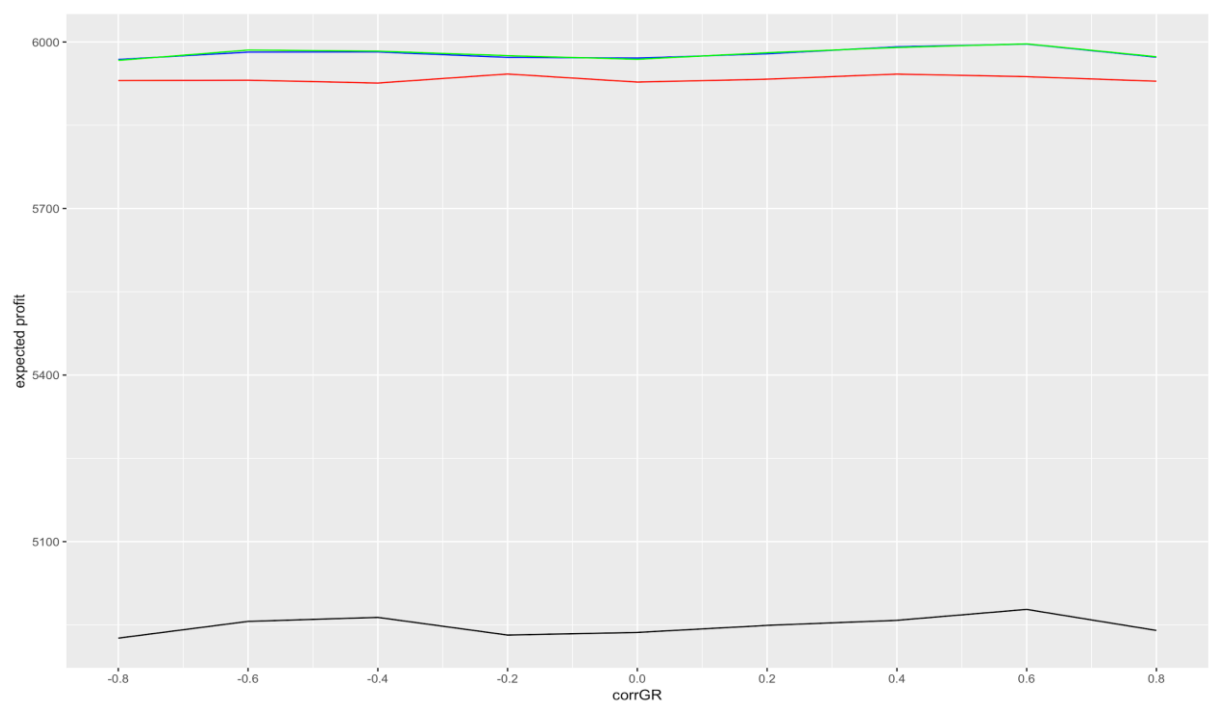
藍: R 船員進入 G 市場, M 船員進入 R 市場

綠: R 船員進入 R 市場, M 船員進入 G 市場

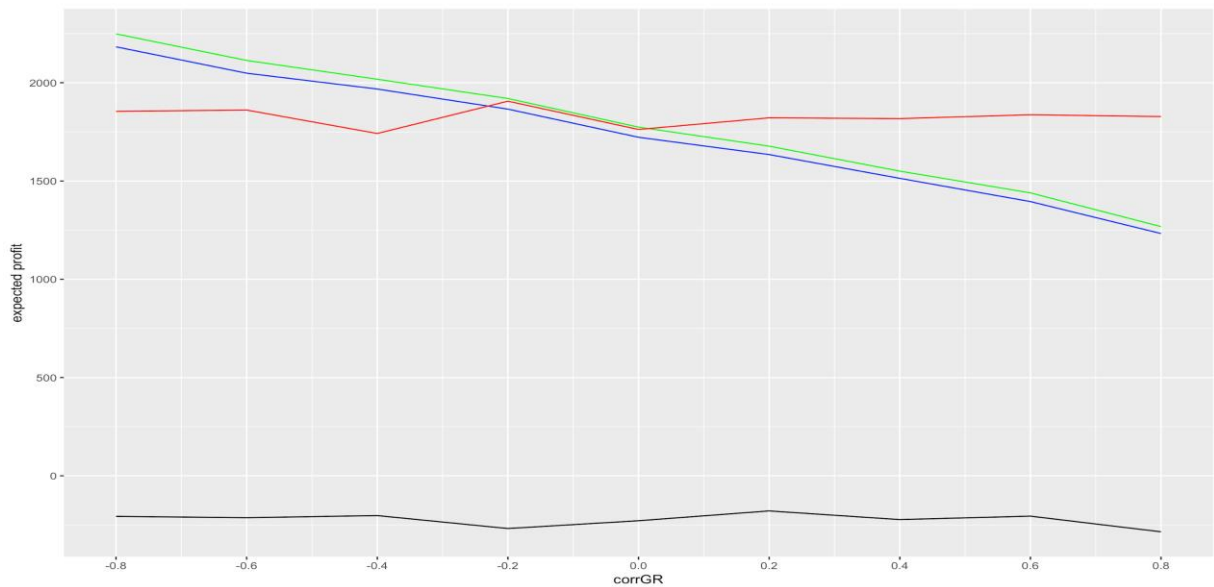
紅: R 船員進入 R 市場, M 船員進入 R 市場

黑: R 船員進入 G 市場, M 船員進入 G 市場

1. y-axis is expected profit and x-axis is corr(PriceGlou, PriceRock)



2. y-axis is CVAR(5%) and x-axis is $\text{corr}(\text{PriceGlou}, \text{PriceRock})$



3. 分別將漁獲分散於兩個市場賣可能造成競爭，價格可能因此下跌，而造成獲利下降。若直接將漁獲集中於一個市場賣，則無競爭者問題，價格領導較有優勢，因此可維持獲利水準。在經濟學中的供給需求曲線，可得知市場需求總值不變，因此圖顯示的線大致為水平。

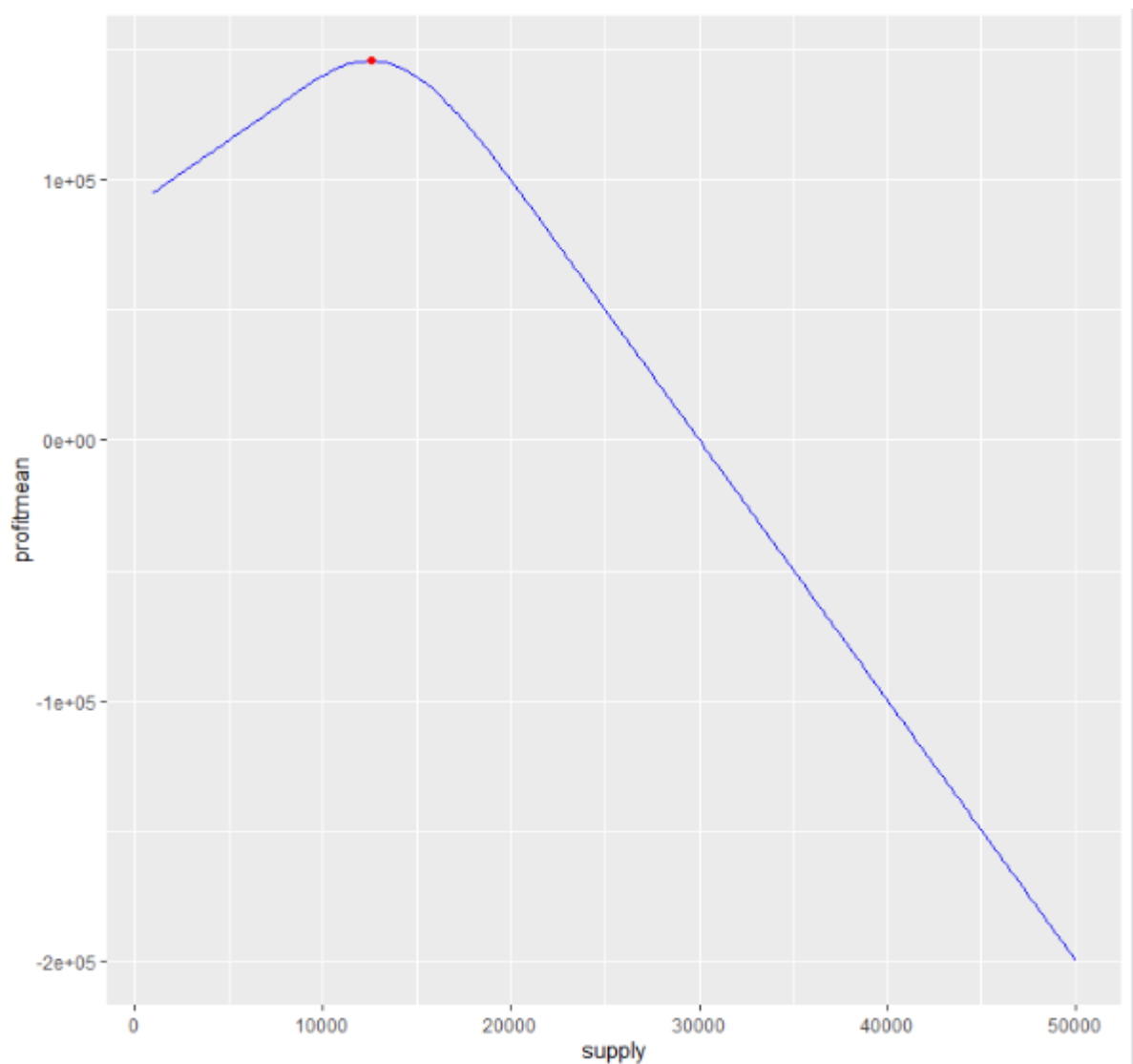
兩市場的相關性越大，越容易削價競爭，造成獲利下降，反之市場相關性低時，產品在某一市場具有獨特性，因此獲利期望值較市場正相關時大。因此在產品進入市場時，公司應進行集中市場策略，所得之獲利最大。

Q2:

```
1 cost=10#每件衣服的價格
2 before_price=25#比賽開打前衣服售價
3 win_price=before_price#雄鷹贏球衣服售價
4 lose_price=12.5#雄鷹輸球衣服售價
5 n=1000#模擬1000場比賽
6
7
8 win_prob=0.4#雄鷹贏球機率
9
10 # mean 2000 and standard deviation of 1000
11 #ex=shape*scale, varx=shape*scale^2
12 scale=(1000^2)/2000#gamma的scale參數計算(var/mean)
13 sh=2000/scale#gamma的shape參數計算(mean/scale)
14
15
16
17 supply=seq(from=1000, to=50000, by=200)#模擬supply(1000~100000, 增量1000為一單位)
18 profit_one<-c()#存放二次模擬的不同supply情況下的所有profit
19 demand_one<-c()
20 profit_table<-data.frame(matrix(ncol = length(supply), nrow = 0))#建立一個空的dataframe, 存放10000次模擬結果
21 game=sample(c(1,0), n, replace=TRUE, prob=c(win_prob, 1-win_prob))#模擬10000次比賽
22 demand_table<-data.frame(matrix(ncol = length(supply), nrow = 0))
23 for (i in c(1:10000)){
24   before_demand=rnorm(1,9000,2000)#比賽前的demand
25   if(game[i]==1){ #贏球
26     demand=rnorm(1,6000,2000)#模擬贏球後球衣需求
27   }
28   } else{ #輸球
29     demand=rgamma(1,shape=sh,scale=scale)#模擬輸球後球衣需求
30   }
31 }
32
33 for(s in supply){#模擬不同supply下之所有獲利情況
34   if(game[i]==1){
35     profit=min(s,before_demand)*(before_price-cost)+max(0,min(s-before_demand,demand))*(win_price-cost)-(s-before_demand-max(0,min(s-before_demand,demand)))*cost
36     #獲利=min(生產量,比賽前的需求)*比賽前單件球衣利潤+max(0,min(比賽後僅存庫存,比賽贏球後需求))-比賽結束後沒賣掉的衣服*成本
37   }
38   } else{
39     profit=min(s,before_demand)*(before_price-cost)+max(0,min(s-before_demand,demand))*(lose_price-cost)-(s-before_demand-max(0,min(s-before_demand,demand)))*cost
40     #獲利=min(生產量,比賽前的需求)*比賽前單件球衣利潤+max(0,min(比賽後僅存庫存,比賽輸球後需求))-比賽結束後沒賣掉的衣服*成本
41   }
42   d=before_demand+demand
43   profit_one=c(profit_one,profit)#加入該需求模擬一次比賽後的獲利
44 }
45
46 total=demand+before_demand#比賽前+比賽後的需求
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

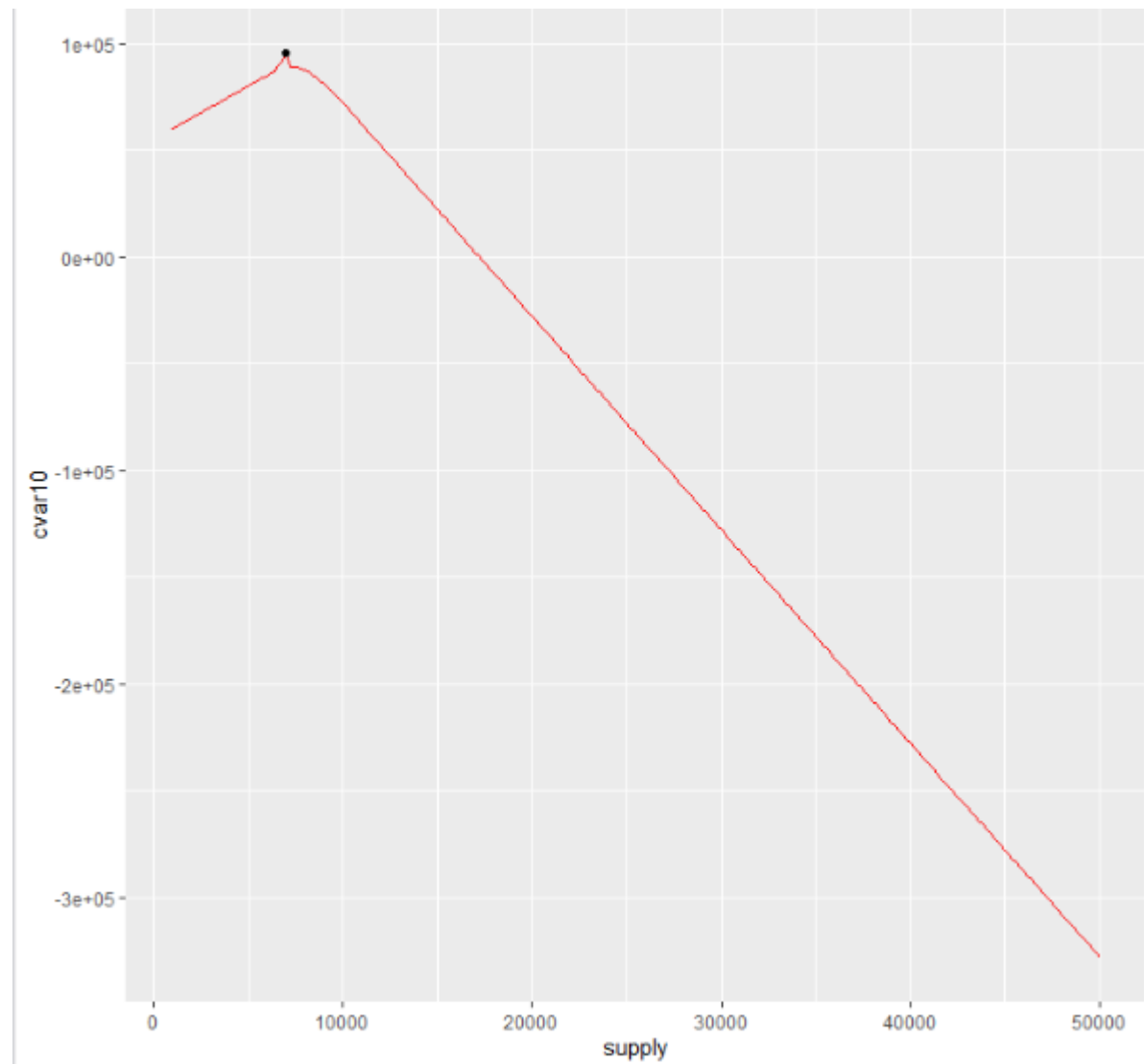
1. 下圖是供應與在題目給定之各種條件下平均獲利的相關圖，紅色點為獲利最大的點，而對應的 SUPPLY 量為 12600。SUPPLY 的模擬區間設 1000~50000 是因為比賽前的需求量加上比賽後的最大需求量(贏球情況)為 $9000+6000=15000$ ，標準差為 2000^2+2000^2 開根號大約=2800。15000 往左六個標準差可以滿足 99.999% 之最低需求(SIX-SIGMA)，又因為往左六個標準差會導致需求變為 <0，因此改為五個標準差($15000-14000=1000$)，因此從 1000 開始跑，又因電腦效能之問題，無法一個一個需求慢慢試，因此以 200 為單位往上遞增到 50000。

optimal_profit of (1): 145787 optimal_supply of (1): 12600



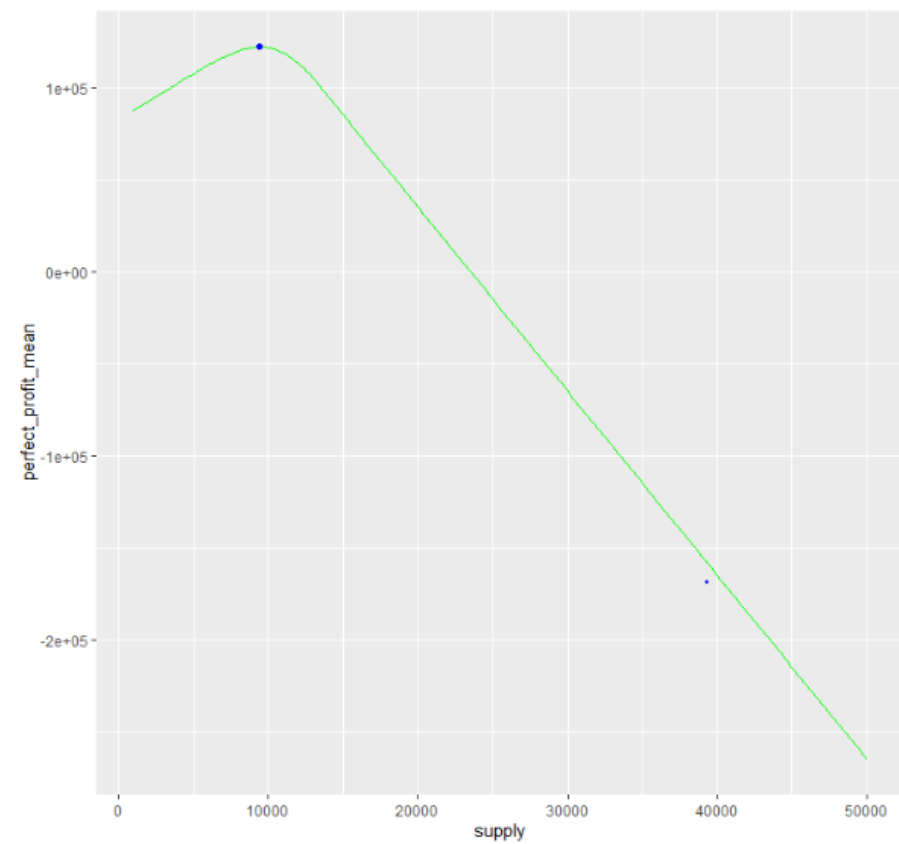
2. 下圖是供應與在題目給定之各種條件下 CVAR10 的平均獲利相關圖，黑色點為獲利最大的點，而對應的 SUPPLY 量為 7000。

optimal_profit of (2): 95295 optimal_supply of (2): 7000



3. 依照報童模型: $CU/CO+CU$ 可以得出需要的完美資訊 $PROB$ 為 0.6 ，因此我們先用先前存下來的 demand table(table 存放各個 supply 的情況下比賽前、比賽後、和總需求)，我們找出總需求 $PROB=0.6$ 的 quantile q 然後取所有各個 supply 小於 q 的所有資料平均，以及供給量進行作圖。下圖是供應與在題目給定之各種條件下，獲利小於 $QUANTILE(prob=0.6)$ 的平均獲利相關圖，藍色點為獲利最大的點，而對應的 $SUPPLY$ 量為 9400。

optimal_profit of (3): 121911 optimal_supply of (3): 9400



Ps.三題的比較圖表:

