Final Examine

711378912 蔡宜諠

2025-06-05

Q1: (30%) Car Insurance Data Analysis

```
library(MASS)
data("Insurance")
str(Insurance)
```

```
## 'data.frame':    64 obs. of  5 variables:
##  $ District: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Group   : Ord.factor w/ 4 levels "<1l"<"1-1.5l"<..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ Age     : Ord.factor w/ 4 levels "<25"<"25-29"<..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ Holders : int  197 264 246 1680 284 536 696 3582 133 286 ...
##  $ Claims  : int  38 35 20 156 63 84 89 400 19 52 ...
```

- District (factor): district of residence of policyholder (1 to 4): 4 is major cities.
- Group (an ordered factor): group of car with levels <1 litre, 1–1.5 litre, 1.5–2 litre, >2 litre.
- Age (an ordered factor): the age of the insured in 4 groups labelled <25, 25–29, 30–35, >35.
- Holders: numbers of policyholders.
- Claims: numbers of claims

請建構一合理模型，用以預測不同地區(District)、不同年齡層(Age)及不同種車輛(Group)對申請理賠數(Claims)的影響。 其中保單持有人數(Holders)會對申請理賠數有重要的影響，因為保單持有人數愈多才有可能有愈多的申請理賠數。請建構二個以上的模型， 並用AIC進行模型選擇。

```
head(Insurance)
```

```
##   District  Group   Age Holders Claims
## 1        1    <1l   <25     197     38
## 2        1    <1l 25-29     264     35
## 3        1    <1l 30-35     246     20
## 4        1    <1l   >35    1680    156
## 5        1 1-1.5l   <25     284     63
## 6        1 1-1.5l 25-29     536     84
```

```
# 1. 確認變數型態與無遺失值
str(Insurance); sum(is.na(Insurance))  # District/Group/Age 都是因子、Holders/Claims 為整數，且無 NA
```

```
## 'data.frame':    64 obs. of  5 variables:
##  $ District: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Group   : Ord.factor w/ 4 levels "<1l"<"1-1.5l"<..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ Age     : Ord.factor w/ 4 levels "<25"<"25-29"<..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ Holders : int  197 264 246 1680 284 536 696 3582 133 286 ...
##  $ Claims  : int  38 35 20 156 63 84 89 400 19 52 ...
## [1] 0
```

```
# 2. 檢查 Poisson 過度離散
dispersion_ratio <- {
  tmp <- glm(Claims ~ District + Age + Group, offset = log(Holders),
             family = poisson, data = Insurance)
  sum(residuals(tmp, type="pearson")^2) / tmp$df.residual
}
dispersion_ratio  # 若 >1.5，直接改用 glm.nb()
```

```
## [1] 0.9005432
```

dispersion_ratio < 1.5 表示並無明顯過度離散，故可直接使用 Poisson 回歸模型。

```
# 開始建立模型为
# 3-1. Poisson 回歸（無交互作用）
fit_pois <- glm(Claims ~ District + Age + Group,
                offset = log(Holders),
                family = poisson(link="log"),
                data = Insurance)
summary(fit_pois)
```

```
## 
## Call:
## glm(formula = Claims ~ District + Age + Group, family = poisson(link = "log"), 
##     data = Insurance, offset = log(Holders))
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.810508   0.032972 -54.910  < 2e-16 ***
## District2    0.025868   0.043016   0.601 0.547597    
## District3    0.038524   0.050512   0.763 0.445657    
## District4    0.234205   0.061673   3.798 0.000146 ***
## Age.L       -0.394432   0.049404  -7.984 1.42e-15 ***
## Age.Q       -0.000355   0.048918  -0.007 0.994210    
## Age.C       -0.016737   0.048478  -0.345 0.729910    
## Group.L      0.429708   0.049459   8.688  < 2e-16 ***
## Group.Q      0.004632   0.041988   0.110 0.912150    
## Group.C     -0.029294   0.033069  -0.886 0.375696    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 236.26  on 63  degrees of freedom
## Residual deviance:  51.42  on 54  degrees of freedom
## AIC: 388.74
## 
## Number of Fisher Scoring iterations: 4
```

```
# 3-2. Negative Binomial 回歸（無交互作用）
fit_nb <- glm.nb(Claims ~ District + Age + Group + offset(log(Holders)),
                 data = Insurance)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
summary(fit_nb)
```

```
##
## Call:
## glm.nb(formula = Claims ~ District + Age + Group + offset(log(Holders)),
##     data = Insurance, init.theta = 449933.5546, link = log)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.8105087  0.0329755 -54.905  < 2e-16 ***
## District2    0.0258703  0.0430236   0.601 0.547637
## District3    0.0385270  0.0505189   0.763 0.445686
## District4    0.2342060  0.0616789   3.797 0.000146 ***
## Age.L       -0.3944313  0.0494064  -7.983 1.42e-15 ***
## Age.Q       -0.0003568  0.0489205  -0.007 0.994181
## Age.C       -0.0167372  0.0484804  -0.345 0.729916
## Group.L      0.4297077  0.0494627   8.688  < 2e-16 ***
## Group.Q      0.0046344  0.0419919   0.110 0.912120
## Group.C     -0.0292999  0.0330738  -0.886 0.375675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(449933.6) family taken to be 1)
##
##     Null deviance: 236.212  on 63  degrees of freedom
## Residual deviance:  51.416  on 54  degrees of freedom
## AIC: 390.74
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  449934
##           Std. Err.:  4185443
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -368.745
```

```
# 3-3. Poisson 加交互作用
fit_pois_int <- glm(Claims ~ District * Group + Age * Group,
                    offset = log(Holders),
                    family = poisson(link="log"),
                    data = Insurance)
summary(fit_pois_int)
```

```
##
## Call:
## glm(formula = Claims ~ District * Group + Age * Group, family = poisson(link = "log"),
##     data = Insurance, offset = log(Holders))
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.829320   0.039004 -46.901  < 2e-16 ***
## District2            0.036678   0.051177   0.717  0.47357
## District3            0.016447   0.060312   0.273  0.78508
## District4            0.266345   0.069617   3.826  0.00013 ***
## Group.L              0.403559   0.092834   4.347 1.38e-05 ***
## Group.Q             -0.011077   0.078008  -0.142  0.88708
## Group.C             -0.019733   0.059603  -0.331  0.74058
## Age.L               -0.373827   0.067105  -5.571 2.54e-08 ***
## Age.Q               -0.012604   0.062699  -0.201  0.84068
## Age.C                0.013476   0.058060   0.232  0.81645
## District2:Group.L -0.061999   0.120616  -0.514  0.60724
## District3:Group.L -0.156616   0.142420  -1.100  0.27147
## District4:Group.L  0.250334   0.161105   1.554  0.12022
## District2:Group.Q  0.067631   0.102353   0.661  0.50876
## District3:Group.Q -0.082433   0.120623  -0.683  0.49436
## District4:Group.Q  0.106839   0.139234   0.767  0.44288
## District2:Group.C -0.005451   0.080026  -0.068  0.94570
## District3:Group.C -0.098784   0.093896  -1.052  0.29277
## District4:Group.C  0.048058   0.113213   0.424  0.67121
## Group.L:Age.L      0.201913   0.160412   1.259  0.20813
## Group.Q:Age.L     -0.115387   0.134210  -0.860  0.38993
## Group.C:Age.L     -0.075131   0.101451  -0.741  0.45896
## Group.L:Age.Q     -0.180135   0.149434  -1.205  0.22803
## Group.Q:Age.Q      0.135675   0.125398   1.082  0.27927
## Group.C:Age.Q      0.160283   0.095492   1.678  0.09325 .
## Group.L:Age.C      0.045273   0.137769   0.329  0.74245
## Group.Q:Age.C      0.062498   0.116121   0.538  0.59043
## Group.C:Age.C      0.011679   0.089374   0.131  0.89603
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.259  on 63  degrees of freedom
## Residual deviance:  33.527  on 36  degrees of freedom
## AIC: 406.85
##
## Number of Fisher Scoring iterations: 5
```

```
# 3-4. Negative Binomial 加交互作用
fit_nb_int <- glm.nb(Claims ~ District * Group + Age * Group + offset(log(Holders)),
                     data = Insurance)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
## Warning in glm.nb(Claims ~ District * Group + Age * Group +
## offset(log(Holders)), : alternation limit reached
```

```
summary(fit_nb_int)
```

```
## 
## Call:
## glm.nb(formula = Claims ~ District * Group + Age * Group + offset(log(Holders)), 
##     data = Insurance, init.theta = 463341.2966, link = log)
## 
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       -1.829321   0.039006 -46.898  < 2e-16 ***
## District2          0.036682   0.051181   0.717  0.47356    
## District3          0.016450   0.060316   0.273  0.78506    
## District4          0.266343   0.069621   3.826  0.00013 ***
## Group.L            0.403558   0.092837   4.347 1.38e-05 ***
## Group.Q           -0.011079   0.078013  -0.142  0.88706    
## Group.C           -0.019724   0.059609  -0.331  0.74074    
## Age.L             -0.373827   0.067106  -5.571 2.54e-08 ***
## Age.Q             -0.012603   0.062700  -0.201  0.84069    
## Age.C              0.013476   0.058062   0.232  0.81647    
## District2:Group.L -0.061996   0.120624  -0.514  0.60728    
## District3:Group.L -0.156613   0.142428  -1.100  0.27151    
## District4:Group.L  0.250338   0.161112   1.554  0.12023    
## District2:Group.Q  0.067641   0.102363   0.661  0.50874    
## District3:Group.Q -0.082437   0.120632  -0.683  0.49437    
## District4:Group.Q  0.106841   0.139241   0.767  0.44290    
## District2:Group.C -0.005478   0.080039  -0.068  0.94544    
## District3:Group.C -0.098796   0.093907  -1.052  0.29277    
## District4:Group.C  0.048046   0.113223   0.424  0.67131    
## Group.L:Age.L      0.201909   0.160415   1.259  0.20815    
## Group.Q:Age.L     -0.115386   0.134213  -0.860  0.38994    
## Group.C:Age.L     -0.075123   0.101455  -0.740  0.45902    
## Group.L:Age.Q     -0.180132   0.149436  -1.205  0.22804    
## Group.Q:Age.Q      0.135675   0.125401   1.082  0.27928    
## Group.C:Age.Q      0.160282   0.095496   1.678  0.09327 .  
## Group.L:Age.C      0.045273   0.137772   0.329  0.74245    
## Group.Q:Age.C      0.062498   0.116124   0.538  0.59044    
## Group.C:Age.C      0.011676   0.089378   0.131  0.89606    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(463340.8) family taken to be 1)
## 
##     Null deviance: 236.213  on 63  degrees of freedom
## Residual deviance:  33.525  on 36  degrees of freedom
## AIC: 408.85
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  463341 
##           Std. Err.:  3657633 
## Warning while fitting theta: alternation limit reached 
## 
##  2 x log-likelihood:  -350.853
```

拿出剛剛的所有結果比較一下下

```
models <- list(
  Poisson_無交互 = fit_pois,
  NB_無交互      = fit_nb,
  Poisson_含交互 = fit_pois_int,
  NB_含交互      = fit_nb_int
)
aic_tbl <- data.frame(
  Model = names(models),
  AIC   = sapply(models, AIC)
)
aic_tbl[order(aic_tbl$AIC), ]
```

```
##                         Model      AIC
## Poisson_無交互 Poisson_無交互 388.7416
## NB_無交互           NB_無交互 390.7450
## Poisson_含交互 Poisson_含交互 406.8486
## NB_含交互           NB_含交互 408.8534
```

Poisson 回歸（無交互作用） AIC 最低是最佳模型。

Q2: (30%) 乒乓球世界排名資料分析

根據國際桌球總會世界排名，王楚欽、林昀儒、林高遠及林詩棟選手的對戰結果如下表。其中Row是勝利、Column是失敗，格子中是對戰結果的次數。

	WangC	LinY	LinG	LinS
WangC	NA	9	5	3
LinY	3	NA	2	0
LinG	3	6	NA	1
LinS	1	8	0	NA

請用 Bradley-Terry model 分析資料，並給出這些選手能力的排序。

```
X <- matrix(0, 6,6)
X[1,] <- c(1, -1, 0, 0, 9, 3)
X[2,] <- c(1, 0, -1, 0, 5, 3)
X[3,] <- c(1, 0, 0, -1, 3, 1)
X[4,] <- c(0, 1, -1, 0, 2, 6)
X[5,] <- c(0, 1, 0, -1, 0, 8)
X[6,] <- c(0, 0, 1, -1, 1, 0)
colnames(X) <- c("WangC", "LinY", "LinG", "LinS", "nij", "nji")
rownames(X) <- NULL
```

```
# 整理一下表格們
players <- c("WangC", "LinY", "LinG", "LinS")

contests <- data.frame(
  player1 = factor(c("WangC", "WangC", "WangC", "LinY", "LinY", "LinG"),
                   levels = players),
  player2 = factor(c("LinY",  "LinG",  "LinS",  "LinG",  "LinS",  "LinS"),
                   levels = players),
  wins1   = X[, "nij"],
  wins2   = X[, "nji"],
  stringsAsFactors = FALSE
)
contests
```

```
##   player1 player2 wins1 wins2
## 1   WangC    LinY     9     3
## 2   WangC    LinG     5     3
## 3   WangC    LinS     3     1
## 4    LinY    LinG     2     6
## 5    LinY    LinS     0     8
## 6    LinG    LinS     1     0
```

```
library(BradleyTerry2)
```

```
## 
## Attaching package: 'BradleyTerry2'
```

```
## The following object is masked from 'package:MASS':
## 
##     glmmPQL
```

```
bt_model <- BTm(
  outcome = cbind(wins1, wins2),
  player1 = player1,
  player2 = player2,
  data    = contests
)

summary(bt_model)
```

```
##
## Call:
## BTm(outcome = cbind(wins1, wins2), player1 = player1, player2 = player2,
##     data = contests)
##
## Coefficients:
##        Estimate Std. Error z value Pr(>|z|)
## ..LinY  -1.6816     0.5853  -2.873  0.00407 **
## ..LinG  -0.3740     0.5883  -0.636  0.52490
## ..LinS  -0.1434     0.7168  -0.200  0.84147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19.2610  on 6  degrees of freedom
## Residual deviance:  6.3111  on 3  degrees of freedom
## AIC: 21.613
##
## Number of Fisher Scoring iterations: 4
```

```
abilities <- BTabilities(bt_model)
sort(abilities[, "ability"], decreasing = TRUE)
```

```
##      WangC       LinS       LinG       LinY
##  0.0000000 -0.1433565 -0.3740310 -1.6816201
```

WangC 能力值最高，其次依序為 LinS、LinG，最後是 LinY。

Q3:(40%)懨食症資料

```
data("anorexia")
str(anorexia)
```

```
## 'data.frame':    72 obs. of  3 variables:
##  $ Treat : Factor w/ 3 levels "CBT","Cont","FT": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Prewt : num  80.7 89.4 91.8 74 78.1 88.3 87.3 75.1 80.6 78.4 ...
##  $ Postwt: num  80.2 80.1 86.4 86.3 76.1 78.1 75.1 86.7 73.5 84.6 ...
```

- Treat (factor of three levels): “Cont” (control), “CBT” (Cognitive Behavioural treatment) and “FT” (family treatment).
- Prewt: weight of patient before study period, in lbs.
- Postwt: weight of patient after study period, in lbs.

假設成年人體重長期少於82lb會對身體造成無法挽回之傷害

```
anorexia$Prewt_B <- ifelse(anorexia$Prewt < 82, 1, 0)
anorexia$Postwt_B <- ifelse(anorexia$Postwt < 82, 1, 0)
```

請用GLMM建構模型分析資料，目標是看看這些治療何者對於避免患者體重少82lb更有用。

```
# 先從建立模型開始!
fit <- glm(Postwt_B ~ Treat + Prewt_B,
           family = binomial,
           data   = anorexia)
# 計算 Wald 近似的 95% CI（不做 profile）
ci_wald <- exp(confint.default(fit))
cbind(OR = exp(coef(fit)), CI_low = ci_wald[,1], CI_high = ci_wald[,2])
```

```
##                    OR     CI_low   CI_high
## (Intercept) 0.1730916 0.05737192  0.522219
## TreatCont   5.2055023 1.53687071 17.631447
## TreatFT     0.9640520 0.22468974  4.136354
## Prewt_B     4.0704320 1.36664975 12.123382
```

1. Cont vs CBT：OR ≈ 5.21（95% CI = [1.54, 17.63]），顯示 Cont 組患者比 CBT 組更容易在治療後仍低於 82 lb。
2. FT vs CBT：OR ≈ 0.96（95% CI = [0.22, 4.14]），未達顯著差異。
3. Prewt_B（治療前 < 82 lb）：OR ≈ 4.07（95% CI = [1.37, 12.12]），表示治療前體重偏低者比體重正常者更容易治療後仍低於 82 lb。