作業二

管科所碩一 / 孫守謙 / 0853132

一、 資料設定、讀取

```
#Load the data
dataset = read.csv("Accidents.csv", header=T, fileEncoding="Big5")
#第 9643 項資料全部都是空的
dataset <- dataset[-9643, ]
```

	編號	發生星期	GPS經度	GPS緯度	天候代碼	天候名稱	光線代碼	光線名稱	路面狀況.路面狀態代碼	路面狀況名稱	當事者性別代碼	當事者性別名稱	當事者年齡	車種代碼	車種名稱	保護裝備代碼	保護裝備名稱	飲酒情形代碼	飲酒情形名稱	事故類別名稱
	<fct>	<fct>	<dbl>	<dbl>	<int>	<fct>	<int>	<fct>	<int>	<fct>	<int>	<fct>	<int>	<fct>	<fct>	<int>	<fct>	<int>	<fct>	<fct>
1	10607BFY00A0001	三	120.9303	24.77982	8	晴	1	日間自然光線	5	乾燥	3	無或物(動物、堆置物)	-1			NA		NA		A1
2	10607BFY00A0003	六	120.9528	24.80693	8	晴	1	日間自然光線	5	乾燥	1	男	44	C0	機車	1	戴安全帽或繫安全帶(使用幼	2	經檢測無酒精反應	A1

可以觀察到這份資料有 9642 筆數據，且總共有 20 個特徵。

二、 資料分析、預處理

- 預測目標 -> 事故類別名稱

事故類別名稱為這組資料集的目標特徵，事故類別分為三種 A1, A2, A3，其中三者分別代表意思為：

- A1 類指造成人員當場或二十四小時內死亡之交通事故
- A2 類指造成人員受傷或超過二十四小時死亡之交通事故
- A3 類指僅有車輛財物受損之交通事故

事故類別	A1	A2	A3
發生次數	13	3596	6033

由於 A1 類事故太少，較難訓練出預測 A1 事故的模型，所以將問題改為：**發生的事故有沒有人員死亡？**

也就是 A1+A2 為一類，而 A3 為一類的分類問題。

```
dataset$事故類別名稱  = gsub('A1',1, dataset$事故類別名稱)
dataset$事故類別名稱  = gsub('A2',1, dataset$事故類別名稱)
dataset$事故類別名稱  = gsub('A3',0, dataset$事故類別名稱)
dataset$事故類別名稱  = as.integer(dataset$事故類別名稱)
```

這邊把 A1,A2 設定為 **1**，並且把 A3 設定為 **0**。
表示有人死亡以及沒有人死亡的布林問題。

- **缺失值處理**

由 colnames(dataset) 來觀察有哪些特徵：

編號、發生星期、GPS 經度、GPS 緯度、天候代碼、天候名稱、光線代碼、光線名稱、路面狀態代碼、路面狀況名稱、當事者性別代碼、當事者性別名稱、當事者年齡、車種代碼、車種名稱、保護裝備代碼、保護裝備名稱、飲酒情形代碼、飲酒情形名稱

這組數據有 19 個特徵。而其中特別的是 *"天候、光線、路面狀況、當事者性別、車種、保護裝備、飲酒情形情形"* 都各有兩個特徵，實際情況敘述以及其情況的代碼編號。所以只需要在代碼與名稱之間留下缺失比較少的那一項就好。

以 sapply(dataset,function(df){sum(is.na(df)/nrow(dataset))}) 來觀察每個特徵的缺失率：

特徵缺失率				
編號	發生星期	GPS 經度	GPS 緯度	天候代碼
0.00	0.00	0.00	0.00	0.1328
天候名稱	光線代碼	光線名稱	路面狀態代碼	路面狀況名稱
0.00	0.6011	0.00	0.6043	0.00
當事者性別代碼	當事者性別名稱	當事者年齡	車種代碼	車種名稱
0.00	0.00	0.00	0.00	0.00
保護裝備代碼	保護裝備名稱	飲酒情形代碼	飲酒情形名稱	
0.5907	0.00	0.5908	0.00	

有幾個缺失率較高的特徵如：*光線代碼、路面狀態代碼、保護裝備代碼、飲酒情形代碼*，
都缺了超過一半的資料，所以我選擇捨棄掉代碼這項特徵，利用他的類別名稱去做處理就好。
而車種與天候名稱與代碼，雖然都很少缺失值，但他們基本上是代表相同意思，所以選擇一個就好。

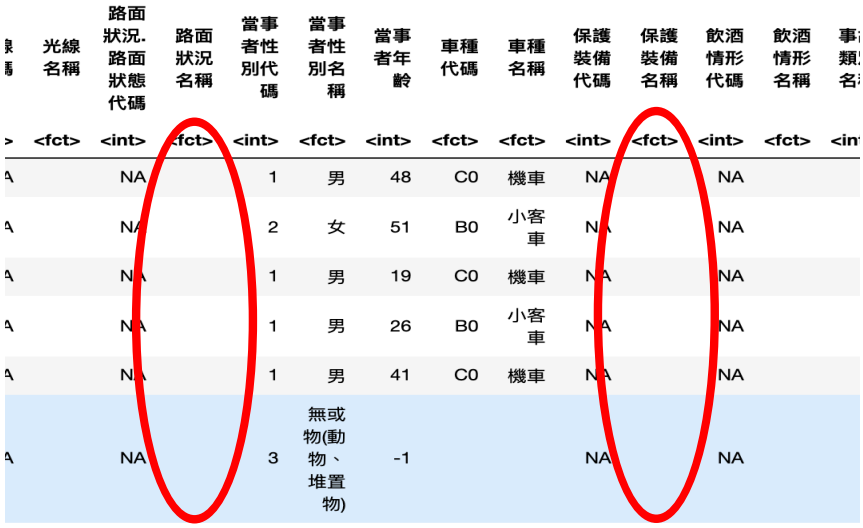
- 天候名稱與天候代碼這組特徵有點問題，有可能天氣是晴朗對應到代號是 8，但另一組數據天氣是暴雨但代碼也依樣是 8，所以把代碼的部分刪除，參考天候名稱而已。

	編號	發生星期	GPS經度	GPS緯度	天候代碼	天候名稱	光線代碼
	<fct>	<fct>	<dbl>	<dbl>	<int>	<fct>	<int>
9637	10612BFY00D0170	日	120.9626	24.81792	8	暴雨	NA
9638	10612BFY00D0171	一	120.9934	24.79206	8	暴雨	NA
1	10607BFY00A0001	三	120.9303	24.77982	8	晴	
2	10607BFY00A0003	六	120.9528	24.80693	8	晴	

```
dataset <- subset(dataset, select = c(-編號,-光線代碼, -路面狀況.路面狀態代碼,-保護裝備代碼,-飲酒情形代碼,-車種代碼,-天候代碼,-當事者性別名稱))
```

若只觀察”缺失率“會以為在刪除一些缺失率較高的特徵後便處理完成，但實際去觀察資料集時會發現，在許多欄位會有 '', ' ' 這樣的空格值，系統並沒有把這些認定成 NA, 所以上面在觀察缺失率的時候才會那麼低。

所以現在必須把那些空格值轉換成 NA 再進行填補 Missing Value 的動作。



光線名稱	路面狀況.路面狀態代碼	路面狀況名稱	當事者性別代碼	當事者性別名稱	當事者年齡	車種代碼	車種名稱	保護裝備代碼	保護裝備名稱	飲酒情形代碼	飲酒情形名稱	事
<fct>	<int>	<fct>	<int>	<fct>	<int>	<fct>	<fct>	<int>	<fct>	<int>	<fct>	<int>
	NA		1	男	48	C0	機車	NA		NA		
	NA		2	女	51	B0	小客車	NA		NA		
	NA		1	男	19	C0	機車	NA		NA		
	NA		1	男	26	B0	小客車	NA		NA		
	NA		1	男	41	C0	機車	NA		NA		
	NA		3	無或物(動物、堆置物)	-1			NA		NA		

```
dataset$天候名稱  <- gsub(' ', NA , dataset$天候名稱)
dataset$光線名稱  <- gsub(' ', NA , dataset$光線名稱)
dataset$車種名稱  <- gsub(' ', NA , dataset$車種名稱)
dataset$保護裝備名稱  <- gsub(' ', NA , dataset$保護裝備名稱)
dataset$飲酒情形名稱  <- gsub(' ', NA , dataset$飲酒情形名稱)
```

更動過後的特徵缺值率					
發生星期	GPS 經度	GPS 緯度	天候名稱	光線名稱	路面狀況名稱
0.00	0.00	0.00	0.0132	0.6012	0.00
當事者性別代碼	當事者年齡	車種名稱	保護裝備名稱	飲酒情形名稱	事故類別名稱
0.00	0.00	0.04874	0.597488	0.5908	0.00

大部分的特徵都已經處理完成，現在僅剩'天候名稱,光線名稱,車種名稱,保護裝備名稱,飲酒情形名稱'這五個特徵需要處理。

- **觀察“光線名稱”**

Var1	Var2	Freq
<fct>	<fct>	<int>
晨或暮光	0	10
日間自然光線	0	188
夜間(或隧道、地下道、涵洞)無照明	0	16
夜間(或隧道、地下道、涵洞)有照明	0	51
晨或暮光	1	44
日間自然光線	1	2590
夜間(或隧道、地下道、涵洞)無照明	1	61
夜間(或隧道、地下道、涵洞)有照明	1	886

這邊觀察“光線名稱”與“事故類別”的關係，可以發現不管是有死亡事故還是無死亡事故，都是“日間自然光線”最多，這有可能導致訓練時，會錯誤地認為自然光線容易造成死亡事故嗎？

“光線名稱”原本就有大概六成的缺失值，再加上剩下的資料有分布不均的問題，較難有效地去處理這項特徵，所以決定直接刪除。

- **觀察“飲酒情形名稱**

Var1	Freq
<fct>	<int>
不明	62
非駕駛人,未檢測	279
經觀察未飲酒	317
經呼氣檢測0.16~0.25mg/L或血液檢測0.031%~0.05%	7
經呼氣檢測0.26~0.40mg/L或血液檢測0.051%~0.08%	4
經呼氣檢測0.41~0.55mg/L或血液檢測0.081%~0.11%	9
經呼氣檢測0.56~0.80mg/L或血液檢測0.111%~0.16%	11
經呼氣檢測超0.80~mg/L或血液檢測超過0.16%	18
經呼氣檢測未超過0.15mg/L或血液檢測未超過0.03%	9
經檢測無酒精反應	3176
無法檢測	53

這邊觀察"飲酒情形名稱"，會發現雖然酒駕在我們的認知中是事故發生的重大因素。

但在這組資料集當中，原本在這一項特徵就缺了將近六成的資料，且在剩下有記錄的資料當中有酒駕紀錄的筆數只佔不到 1%，所以這組資料所要探討的主題可能不是在此，所以最終決定把這項特徵給刪除。

- **觀察“保護裝備名稱”**

Var1	Var2	Freq
<fct>	<fct>	<int>
不明	0	2042
戴安全帽或繫安全帶(使用幼童安全椅)	0	441
其他(行人、慢車駕駛人)	0	77
未戴案全帽或未繫安全帶(未使用幼童安全椅)	0	1
不明	1	1076
戴安全帽或繫安全帶(使用幼童安全椅)	1	249
其他(行人、慢車駕駛人)	1	60
未戴案全帽或未繫安全帶(未使用幼童安全椅)	1	0

"保護裝備名稱"與"飲酒情形"的狀況相近，在原本就缺了將近一半的情況下，在有紀錄的資料裡中欄位又佔了絕大多數。

且觀察"保護裝備名稱"與"事故類別名稱"，並沒有要陳述沒有保護裝備所遇到的意外會比較嚴重的狀況，所以也將此特徵刪除。

- **剩下的“車種名稱”與“天候名稱”，因缺失程度較小，所以直接把有缺失的資料刪除，此時便處理完缺失值的問題。**

- **設置 Dummy Variable**

在處理完缺失值後，僅存的只需要把剩下的變數轉變為虛擬變數。除了當事者年齡、GPS 經度、GPS 緯度為數值變數之外，其他的變數都需要做虛擬變數的轉換。

```
for(unique_value in unique(dataset$天候名稱)){dataset[paste("Wether", unique_value, sep = ".")] <- ifelse(dataset$天候名稱== unique_value, 1, 0)}
for(unique_value in unique(dataset$路面狀況名稱)){dataset[paste("Roadtype", unique_value, sep = ".")] <- ifelse(dataset$路面狀況名稱== unique_value, 1, 0)}
for(unique_value in unique(dataset$當事者性別代碼)){dataset[paste("Gender", unique_value, sep = ".")] <- ifelse(dataset$當事者性別代碼== unique_value, 1, 0)}
for(unique_value in unique(dataset$車種名稱)){dataset[paste("Transportation", unique_value, sep = ".")] <- ifelse(dataset$車種名稱== unique_value, 1, 0)}
for(unique_value in unique(dataset$發生星期)){dataset[paste("DayOfWeek", unique_value, sep = ".")] <- ifelse(dataset$發生星期== unique_value, 1, 0)}
```

Wether.晴	Wether.陰	Wether.雨	Wether.暴雨	Wether.強風	Wether.霧或煙	…	Transportation.特種車	Transportation.曳引車
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	…	<dbl>	<dbl>
1	0	0	0	0	0	…	0	0
1	0	0	0	0	0	…	0	0
1	0	0	0	0	0	…	0	0
0	1	0	0	0	0	…	0	0
1	0	0	0	0	0	…	0	0
1	0	0	0	0	0	…	0	0

在處理完後會如上圖，以 0 與 1 表示有沒有發生此情況，到這邊資料預處理就告一段落，可以開始進行預測了。

三、 模型設定與訓練

- **切割訓練、測試資料，比例為 9:1**

```
#Split the data
n <- 0.1*nrow(dataset)
index <- sample(1:nrow(dataset),n)
training <- dataset[-index,]
testing <- dataset[index,]
```

- **模型設定（Logistic Regression）**

```
model <- glm(training$事故類別名稱 ~ ., family = binomial, data=training, control = list(maxit=20)
```

- **預測結果**

```
p2 <- predict(model, testing, type="response")
predict_results <- ifelse(p2 > 0.5, 1, 0)
```

3837: 0 3086: 0 8733: 0 5389: 0 121: 1 2155: 0 1148: 1 7117: 0 8747: 0 3497: 1 9055: 0 7134:
0 2717: 1 9397: 0 6170: 0 5215: 1 4845: 1 1510: 0 6741: 1 6865: 0 1112: 0 9349: 0 7868: 1
6324: 1 8390: 1 8229: 1 9446: 0 1659: 1 6873: 0 9300: 0 1104: 0 8185: 1 8841: 0 9346: 0
2498: 0 6653: 1 7265: 0 8232: 1 2579: 0 7568: 0 4547: 0 3543: 1 1090: 0 3272: 1 3861: 0
8500: 1 5077: 1 8722: 0 7385: 0 6955: 0 7452: 0 8541: 0 3331: 1 105: 1 9620: 0 950: 0 6617: 1
8085: 1 8876: 0 1085: 0 9505: 0 9536: 0 439: 1 101: 1 427: 1 4519: 0 77: 1 1544: 0 4795: 1
2431: 0 9454: 0 5535: 0 8703: 0 8673: 0 3628: 1 206: 1 2524: 0 4061: 0 5992: 0 8791: 0 6259:
1 1033: 0 2020: 1 6735: 1 2771: 0 1247: 0 9576: 0 8112: 1 6292: 1 3125: 0 4355: 0 5721: 0
6842: 1 7776: 1 2909: 0 6860: 1 1452: 0 8916: 0 2347: 0 5600: 0 4612: 0 8657: 0 9459: 0
6888: 0 4218: 0 6654: 1 433: 1 6643: 1 2393: 0 9323: 0 8206: 1 8940: 0 7884: 1 4291: 0 9392:
0 1131: 0 5088: 1 123: 1 8610: 0 8131: 1 4796: 1 4112: 0 2017: 1 1194: 0 6154: 0 6776: 1
5454: 0 1221: 0 6915: 0 8419: 1 6453: 1 3175: 1 531: 1 5846: 0 9188: 0 8909: 0 6201: 1 36: 1
4230: 0 6705: 1 3759: 0 9328: 0 8122: 1 7707: 0 2521: 0 5239: 1 1874: 1 1350: 0 6848: 1
7885: 1 9378: 0 2790: 0 8634: 0 9006: 0 6517: 1 5976: 0 9419: 0 326: 1 7152: 0 2501: 0 966: 0
7511: 0 5484: 0 7781: 0 7199: 0 3601: 1 8210: 1 3062: 0 3016: 1 2199: 0 8226: 1 2718: 0
4311: 0 6135: 0 2538: 0 4966: 1 2943: 0 2288: 0 4787: 1 5000: 1 7769: 0 1007: 0 4864: 1
9270: 0 5548: 0 8936: 0 2453: 0 9482: 0 4790: 1 4063: 0 1285: 0 2558: 0 6033: 0 2000: 1
3467: 1 7391: 0 4510: 0 9054: 0 6113: 0 655: 1 187: … 6241: 1 8751: 1 9636: 0 7876: 1 3825:
1 1815: 1 1964: 1 400: 0 9219: 0 1551: 0 9418: 0 1869: 0 3845: 0 4151: 1 3384: 1 982: 1 5310:
0 7794: 0 2923: 0 3707: 0 3652: 1 6180: 0 9010: 1 6995: 0 6220: 1 6317: 1 1644: 1 3445: 0
6473: 0 8449: 0 2731: 0 8247: 0 3773: 0 3860: 1 316: 1 5296: 0 8156: 1 4272: 0 935: 0 9253: 1
7762: 0 9121: 1 6400: 0 7932: 1 7977: 0 7281: 0 7946: 0 6887: 0 8503: 0 7459: 0 1051: 0
3455: 0 8604: 1 5440: 1 5145: 1 2104: 0 4153: 1 7262: 0 9067: 0 637: 1 5147: 0 863: 0 9139: 0
463: 1 992: 0 3424: 1 8469: 1 8930: 0 9046: 0 7657: 1 2591: 0 8978: 0 5061: 0 557: 1 1884: 1
2481: 0 2341: 0 5643: 1 3452: 1 8288: 1 5299: 0 5488: 0 2081: 1 8279: 0 2159: 1 3234: 1
3263: 1 5140: 0 3338: 1 3519: 1 2441: 0 3356: 1 2012: 0 5126: 1 333: 1 7757: 1 978: 1 3360: 0
1279: 0 3913: 0 9208: 1 6935: 1 5538: 0 6966: 1 2307: 1 6861: 1 3428: 1 8189: 0 9198: 0
8744: 1 5033: 1 1904: 0 9150: 1 7606: 0 9132: 0 2372: 0 4336: 0 1709: 0 1168: 1 8211: 1
2880: 0 577: 0 2794: 0 6768: 1 7902: 0 5187: 0 5706: 0 545: 0 3729: 1 6313: 0 7618: 0 2009: 1
4187: 1 7825: 1 7318: 1 3579: 0 4180: 1 7900: 0 9036: 0 9082: 0 2439: 0 7244: 0 5293: 0 268:
1 5700: 0 1488: 0 3076: 0 3442: 0 5174: 0 5275: 0 872: 1 5360: 0 7910: 1 402: 1 1575: 0 3661:
0 4979: 1 9491: 0 91: 0 6864: 0 7666: 1 9360: 1 157: 0 9038: 0 7886: 1 8316: 0 6828: 0 2550: 1

上圖為預測的結果，我們進一步觀察準確率：

```
Accuracy <- mean(predict_results == testing$事故類別名稱)
table(testing$事故類別名稱, predict_results)
```

預測結果混淆矩陣		
	0	1
0	538	15
1	1	351
Accuracy = 0.9823		

結論：

雖然這組資料集相當的凌亂，有很大量的缺失值，且在某些數值有疑似錯誤的情形發生。而資料集當中的某些特徵所表現出的狀況與我們日常生活中的認知較不同，可能就是資料蒐集的方式所導致。但經過簡單的處理之後，最後預測出來的準確率高達 98%，所以這個模型幾乎可以準確的分類發生的意外事件有沒有人員傷亡事件。