MATH3029/4082: Binomial GLMs in R

The dataset considered here relates to the survival of 34 leukemia patients as a function of their white cell bloood count and the presence or absence of a certain characteristic in the cells. The characteristic, when present, is referred to as AG-positive, and when absent, is referred to as AG-negative.

- In the dataset there are 10 groups, each one corresponding to a different combination of AG status and white blood cell count, with the following variables: n_i is the number of patients in group i; y_i is the number of patients in group i who survived at least 52 weeks beyond the time of diagnosis; x_i is the (common) cell count for patients in group i; and $ag_i = 1$ if AG is present in group i, and $ag_i = 2$ if AG is absent in group i. First of all we shall construct the dataset in R.

```
n=c(2,6,3,3,3,1,5,3,6,2)
y=c(1,2,1,1,0,1,3,2,3,1)
ag=c(rep(1,5),rep(2,5))
x=rep(c(500,5000,10000,40000,100000),2)
```

  We now put the data into a 10×4 matrix called `blooddata` and then print it out.

```
blooddata=cbind(n,y,ag,x)
blooddata
```

- **Important**: The structure of the dataset suggests the model $y_i \overset{ind}{\sim} Bin(n_i, p_i), i = 1, \ldots, 10$. (10 rows in the matrix corresponding to unique combinations of predictors), where p_i is the probability that the people in the ith category/row survive for atleast 52 weeks.

- To fit the binomial regression we first need to put y and $n - y$ into a response matrix (Read help file in `R` for `glm`), called $y2$, with two columns, the first column consisting of the "successes", y_i, and the second column consisting of the "failures", $n_i - y_i$. We also need to declare ag as a factor. (Failure to do so will result in `R` perhaps considering as a numeric variable–careful!)

```
y2=cbind(y,n-y)
ag=factor(ag)
```

- The scale of `x` is too large. Let's take a log transform of `x`. This has nothing to do with the link function! We are just transforming a predictor to create a new one.

  Now fit a model with the same slope for both AG groups but different intercepts; call this model M1. Since AG has two levels, the model implicitly creates 1 indicator variable.

```
out1 = glm(y2 ~ ag + log(x), family=binomial(link="logit"))
```

  The logit link is the default in `R`. Other link functions can be used (read help page). The covariate `ag` is included in the model as an indicator variable $z = 0$ if `ag` is 1 and $z = 1$ otherwise. As with a linear model this results in two models which differ only in the intercept term:

$$\begin{aligned} \log(p_i/(1-p_i)) &= \beta_0 + \beta_2 \log(x_i) \quad (\text{when } z = 0) \\ &= \beta_0 + \beta_1 + \beta_2 \log(x_i) \quad (\text{when } z = 1) \end{aligned} \tag{1}$$

The base model corresponds to $ag = 1$, that is AG is present or AG-positive. Thus the model for $z = 1$ (AG-negative) is characterised by a change in intercept of the model with $z = 0$; there is no change in slope (i.e. β_2).

Look at result of fitting with

`summary(out1)`

and note make sure you are comfortable interpreting estimates b_0, b_1 and b_2 of β_0, β_1 and β_2 respectively. For example, $b_0 = 2.4547$ implies that the log-odds of a patient with AG-positive surviving at least 52 weeks beyond the time of diagnosis is 2.4547; and hence, the corresponding probability for the same is $\frac{e^{2.4547}}{1+e^{2.4547}} = 0.92$.

- Is there a higher chance of survival for patients with AG-positive?
  The estimate of the coefficient for `ag` is 1.4408. This implies that the log odds of survival for patients with `ag`=2 is higher by 1.44 than that of patients with `ag`=1. So the answer is no.
- What is the predicted probability of a patient with `ag` 1 and cell count 40000?
  Based on output, since `ag` is 1, the corresponding indicator variable is 0. The estimated linear predictor $\mathbf{x}^T\mathbf{b} = 2.454 + 0 - 0.3664(\log(40000))$. Then $\hat{p} = e^{\mathbf{x}^T\mathbf{b}}/[1 + e^{\mathbf{x}^T\mathbf{b}}]$.
- Recall that the variance matrix of the estimate $\mathbf{b}$ is the inverse of the Fisher Information matrix. For example for the model `out1`, this can be accessed as

  `vcov(out1)`

  Make sure you can match the std error in the output table (from `summary` command) to the diagonal elements of the (observed) FI matrix; recall that the diagonal elements correspond to variances of components of $\mathbf{b}$ while the off-diagonals correspond to the (pairwise) covariances.
- By modifying the model formula in `R`, fit the following models:
  - Model M2: model with only `log(x)`.
  - Model M3: different slopes and different intercepts for both groups (model with `ag`,$\log(x)$ and interaction `ag`*$\log(x)$). Write this down carefully, and carefully interpret the coefficients as in (1) above.

  1. Write down the nesting structure of the models M1, M2, M3.
  2. Perform a Deviance test to check if `ag` is significant at 5%. Does the result of the test corroborate result of the Wald's test provided in the output?
     Let D2 be the *residual deviance* for M2 and D1 for M1. Compare $\Delta = D2 - D1$ against the 95th percentile of a χ^2 distribution with 1 df. [`qchisq`(0.95,1) in `R` for the percentile].
     The p-value in the output against `ag` is the p-value for the large sample Wald's test based on a Z statistic (refer to notes).
  3. Try fitting a binomial GLM with probit and (complementary) log-log link functions, and see how your results change (read the help page for link functions).