

Generalized Linear Models

We have previously worked with regression models where the response variable is quantitative and normally distributed. Now we turn our attention to two types of models where the response variable is discrete and the error terms do not follow a normal distribution, namely logistic regression and Poisson regression. Both belong to a family of regression models called *generalized linear models*.

Generalized linear models are extensions of traditional regression models that allow the mean to depend on the explanatory variables through a link function, and the response variable to be any member of a set of distributions called the exponential family (e.g., Normal, Poisson, Binomial).

We can use the function `glm()` to work with generalized linear models in R. Its usage is similar to that of the function `lm()` which we previously used for multiple linear regression. The main difference is that we need to include an additional argument *family* to describe the error distribution and link function to be used in the model. In this tutorial we show how `glm()` can be used to fit logistic regression and Poisson regression models.

A. Logistic Regression

Logistic regression is appropriate when the response variable is categorical with two possible outcomes (i.e., binary outcomes). Binary variables can be represented using an indicator variable Y_i , taking on values 0 or 1, and modeled using a binomial distribution with probability $P(Y_i=1) = \pi_i$. Logistic regression models this probability as a function of one or more explanatory variables.

To perform logistic regression in R, use the command:

```
> glm(response ~ explanantory_variables, family=binomial)
```

Note that the option *family* is set to *binomial*, which tells R to perform logistic regression.

Ex. A car manufacturer was interested in creating a model for determining the probability that families will purchase a new car during the next year. A random sample of 33 suburban families was selected. Data on annual income (in thousands of dollars) and the current age of the oldest family car (in years) was obtained. A follow-up interview was conducted a year later to determine whether or not the family actually purchased a new car during the year ($Y=1$ if the family purchased a car and 0 otherwise).

We are interested in determining the probability that a family purchases a new car given their income and the age of their oldest car.

To read in the data set and fit a logistic regression model we type:

```
> dat = read.table("Purchase.txt",header=TRUE)
> results = glm(new ~ income + age, family=binomial)
> results
```

```
Call: glm(formula = new ~ income + age, family = binomial)
```

```
Coefficients:
```

```
(Intercept)  income      age
-4.73931     0.06773     0.59863
```

```
Degrees of Freedom: 32 Total (i.e. Null); 30 Residual
```

```
Null Deviance: 44.99
```

```
Residual Deviance: 36.69    AIC: 42.69
```

According to the output, the model is $\text{logit}(\pi_i) = -4.74 + 0.068 \cdot \text{income} + 0.60 \cdot \text{age}$.

After fitting the model, we can test the overall model fit and hypothesis regarding a subset of regression parameters using a likelihood ratio test (LRT). Likelihood ratio tests are similar to partial F-tests in the sense that they compare the full model with a restricted model where the explanatory variables of interest are omitted. The p-values of the tests are calculated using the χ^2 distribution.

To test the hypothesis $H_0: \beta_1 = \beta_2 = 0$ we can compare our model with a reduced model that only contains an intercept term. A likelihood ratio test comparing the full and reduced models can be performed using the `anova()` function with the additional option `test="Chisq"`.

```
> results.reduced = glm(new ~ 1, family=binomial)
> anova(results.reduced, results, test="Chisq")
Analysis of Deviance Table
Model 1: new ~ 1
Model 2: new ~ income + age
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1      32      44.987
2      30      36.690    2    8.298    0.016
```

The likelihood ratio test statistic is $\chi^2=8.298$ with a p-value=0.016. Hence, we have relatively strong evidence in favor of rejecting H_0 .

As a next step, we perform tests on the individual regression parameters.

```
> summary(results)
```

```
Call:
```

```
glm(formula = new ~ income + age, family = binomial)
```

```
Deviance Residuals:
```

```
  Min      1Q  Median      3Q      Max
-1.6189 -0.8949 -0.5880  0.9653  2.0846
```

```
Coefficients:
```

```
            Estimate Std. Error  z value Pr(>|z|)
(Intercept) -4.73931    2.10195   -2.255  0.0242 *
income       0.06773    0.02806    2.414  0.0158 *
age          0.59863    0.39007    1.535  0.1249
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Null deviance: 44.987  on 32  degrees of freedom
```

```
Residual deviance: 36.690  on 30  degrees of freedom
```

```
AIC: 42.69
```

```
Number of Fisher Scoring iterations: 4
```

To test $H_0: \beta_1=0$, we use $z = 2.414$ (p -value=0.0158). Hence, the family's income appears to have a significant impact on the probability of purchasing a new car, while controlling for the age of the families oldest car.

To test $H_0: \beta_2=0$, we use $z = 1.535$ (p -value=0.1249). Hence, the age of a family's oldest car does not appear to have a significant impact on the probability of purchasing a new car, once income is included in the model.

To compute how the odds of purchasing a car changes as a function of income use the commands:

```
> exp(coef(results))
```

```
(Intercept)  income      age
0.008744682  1.070079093  1.819627221
```

To create a 95% confidence interval for the estimate, type:

```
> exp(confint.default(results))
```

```
                2.5 %      97.5 %
(Intercept)  0.0001420897  0.5381773
income       1.0128238514  1.1305710
age          0.8471457285  3.9084695
```

We see that the odds ratio corresponding to income is 1.070 (95% CI: (1.013, 1.131)). This implies that if we fix the age of the oldest car, increasing family income by one thousand dollars will increase the odds of purchasing a new car by 0.07.

We are often interested in using the fitted logistic regression curve to estimate probabilities and construct confidence intervals for these estimates. We can do this using the function `predict.glm`. The usage is similar to that of the function `predict` which we previously used when working on multiple linear regression problems. The main difference is the option `type`, which tells R which type of prediction is required. The default predictions are given on the logit scale (i.e. predictions are made in terms of the log odds), while using `type = "response"` gives the predicted probabilities.

To predict the probability that a family with an annual income of \$53 thousand and whose oldest car is 1 year old will purchase a new car in the next year, type:

```
> pi.hat = predict.glm(results, data.frame(income=53, age=1),  
                        type="response", se.fit=TRUE)  
> pi.hat$fit  
[1] 0.3656668
```

This tells us that the predicted probability is 0.37. In order to obtain confidence intervals we instead need to work on the logit scale and thereafter transform the results into probabilities. To create a 95% confidence interval for the estimate, type:

```
> l.hat = predict.glm(results, data.frame(income=53, age=1), se.fit=TRUE)  
> ci = c(l.hat$fit - 1.96*l.hat$se.fit, l.hat$fit + 1.96*l.hat$se.fit)
```

To transform the results to probabilities type:

```
> exp(ci)/(1+exp(ci))  
[1] 0.1145063 0.7198689
```

For a family with an annual income of \$53 thousand and whose oldest car is 1 year old, the estimated probability of purchasing a new car is 0.366. A 95% CI is given by (0.115, 0.720).

B. Poisson Regression

Data is often collected in counts (e.g. the number of heads in 12 flips of a coin or the number of car thefts in a city during a year). Many discrete response variables have counts as possible outcomes. Binomial counts are the number of successes in a fixed number of trials, n . Poisson counts are the number of occurrences of some event in a certain interval of time (or space). While Binomial counts only take values between 0 and n , Poisson counts have no upper bound.

We now consider a nonlinear regression model where the response outcomes are discrete counts that follow a Poisson distribution. Poisson regression provides a model that describes how the mean response μ , changes as a function of one or more explanatory variables. To perform logistic regression in R, we use the command:

```
> glm(response ~ explanantory_variables, family=poisson)
```

Note that we specified the family to be poisson, which tells R to perform Poisson regression.

Ex. Researchers studied 41 male African elephants over a period of 8 years. The age of the elephant at the beginning of the study and the number of successful matings during the 8 years were recorded. We assume the number of matings follows a Poisson distribution, where the mean depends on the age of the elephant in question.

We can fit a Poisson regression model using the following code:

```
> dat = read.table("elephants.txt", header=TRUE)
> attach(dat)
> results = glm(mating ~ age, family=poisson)
> summary(results)
```

Call:

```
glm(formula = mating ~ age, family = poisson)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.58201	0.54462	-2.905	0.00368 **
age	0.06869	0.01375	4.997	5.81e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 75.372 on 40 degrees of freedom

Residual deviance: 51.012 on 39 degrees of freedom
AIC: 156.46

Number of Fisher Scoring iterations: 5

To determine whether there is a significant relationship between the mean number of matings and the age of the elephants we test $H_0: \beta_1=0$. The test statistic is $z=4.997$ ($p\text{-value}<0.0001$). Hence, it appears that age does impact the mean number of matings.

To see how the mean number of matings increases per year use the commands:

```
> beta =coef(results)
> beta
(Intercept)    age
-1.58200796  0.06869281
> exp(beta[2])
age
1.071107
```

To create a 95% confidence interval for the estimate, type:

```
> exp(confint.default(results))
                2.5 %    97.5 %
(Intercept) 0.07069036 0.5977577
age         1.04263544 1.1003563
```

Hence, each additional year is associated with a 7.1% increase in the mean number of matings. A 95% confidence interval is given by (1.043, 1.100), which represents a 4.3 - 10.0% increase.