GLMs and GAMs

Gavin Simpson

February 2017

Summary

In this lab you will be introduced to the fitting generalised linear models and generalised additive models using R.

1 *Darlingtonia*: Logistic Regression

The first type of GLM we'll consider is a binomial GLM to fit a logistic regression model. We will use this model to test the hypothesis that the probability of wasp visitation at a leaf is related to the height of the left above the ground on specimens of the Cobra Lily (*Darlingtonia californica*.

Begin by loading the data in the data frame `wasp` and convert the variable `visited` to a logical (`TRUE` or `FALSE`)

```
> ## read data, skip 1 as first row contains a comment
> wasp <- read.csv("darlingtonia.csv", skip = 1)
> wasp <- transform(wasp, visited = as.logical(visited))
```

A simple summary of the data is given by a contingency table

```
> with(wasp, table(visited))
```

```
visited
FALSE  TRUE
   32    10
```

Q and A

1. How many observations in the data represent wasp visitations?
2. How many observations in total are there? (Hint: we want to get the number of rows using `nrow()`)

Produce a plot of the data

```
> plot(visited ~ leafHeight, data = wasp)
```

An alternative visualisation can be achieved by plotting kernel density estimates of the distribution of leaf heights for visited and un-visited leaves. For this we'll use the ggplot2 package. Load the package

```
> library("ggplot2")
```

The plot can produced using

```
> plt <- ggplot(wasp, aes(x = leafHeight, colour = visited)) +
+     geom_density()
> plt
```

Q and A

The kernel density estimates are a smooth representation of the density of the data at values of leaf height. These are a smoother version of a histogram and give an approximation of the distribution of values in a data set.

1. Describe the pattern shown in the figure you just drew.
2. What does this plot suggest in terms of our ability to discriminate or predict which leaves will be visited by wasps given their height above the ground?

We'll now fit the logistic regression (binomial GLM with logit link function)

```
> ## fit a logistic regression
> mod <- glm(visited ~ leafHeight, data = wasp, family = binomial)
```

A likelihood ratio test can be performed using the `anova()` method

```
> anova(mod, test = "LRT")

Analysis of Deviance Table

Model: binomial, link: logit

Response: visited

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                          41     46.105
leafHeight  1   19.142        40     26.963 1.213e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Q and A

Using the likelihood ratio test output, answer the following question:

1. Does wasp visitation of leaves depend on the height of the leaf?

The full model summary can be produced using the `summary()` method

```
> ## summary
> summary(mod)

Call:
glm(formula = visited ~ leafHeight, family = binomial, data = wasp)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.18274  -0.46820  -0.23897  -0.08519   1.90573

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.29295    2.16081  -3.375 0.000738 ***
leafHeight   0.11540    0.03655   3.158 0.001591 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 46.105  on 41  degrees of freedom
Residual deviance: 26.963  on 40  degrees of freedom
AIC: 30.963

Number of Fisher Scoring iterations: 6
```

Q and A

Using the likelihood ratio test output, answer the following question:

1. Does wasp visitation of leaves depend on the height of the leaf?

Next we'll produce a plot of the fitted response or model. We do this by predicting from the model for 100 equally spaced values that cover the range of `leafHeight`. The `predict()` function is used, and because we want predictions on the probability (0–1) scale we use `type = "response"`. The final two lines show an alternative way to plot the observations, as rug plots on the upper and lower margins of the figure.

```
> ## predict for 100 equally spaced values
> newD <- with(wasp, data.frame(leafHeight = seq(min(leafHeight),
+                                max(leafHeight), length = 100)))
> pred <- predict(mod, newdata = newD, type = "response")
> ## plot
> plot(pred ~ leafHeight, data = newD, type = "l", ylim = c(0,1),
+      ylab = "Probability of visitation",
+      xlab = "Leaf height (cm)")
> with(wasp, rug(leafHeight[visited == TRUE], side = 3, lwd = 1))
> with(wasp, rug(leafHeight[visited == FALSE], side = 1, lwd = 1))
```

Leave that plot for the moment. It would be useful to have confidence intervals on the fitted curve. We follow a similar recipe as before, predicting for the 100 new observations of `leafHeight`, but this time we want predictions on the scale of η, the linear predictor. The code below does the following things

1. We store the inverse of the link function in `ilogit`; we'll use this to map from the scale of η back on the response scale later
2. We predict for the new observations, this time using `type = "link"`, so the inverse link function hasn't been applied yet. We also ask for the standard errors of the predicted values, again on the scale of η
3. `alpha` is our confidence limit ($1 - \alpha$, really)
4. `crit` is the critical value of the t distribution which we'll use as the multiple to scale the standard error by. This will be close to 1.96 for reasonably sized data sets
5. Next, we convert the predicted values using `ilogit` on to the probability scale of the response, and create the confidence interval, mapping those on to the response scale as well
6. The final two lines of code just add the upper and lower intervals to the plot you made earlier.

```
> ilogit <- family(mod)$linkinv
> predl <- predict(mod, newdata = newD, type = "link", se.fit = TRUE)
> alpha <- 0.05
> crit <- qt(1 - (alpha/2), df = mod$df.residual)
> pred2 <- with(predl,
+               data.frame(fitted = ilogit(fit),
+                          upper = ilogit(fit + (crit * se.fit)),
+                          lower = ilogit(fit - (crit * se.fit)),
+                          leafHeight = newD$leafHeight))
> ## plot
> lines(upper ~ leafHeight, data = pred2, lty = "dashed")
> lines(lower ~ leafHeight, data = pred2, lty = "dashed")
```

2 Galapágos species richness: Poisson GLM

Now we'll look at the Galapágos species richness data. Load the data and show the first few lines of data

```
> gala <- read.csv("galapagos.csv")
> head(gala)
```

```
             Species Endemics  Area Elevation Nearest Scruz Adjacent
Baltra            58       23 25.09       346     0.6   0.6     1.84
Bartolome         31       21  1.24       109     0.6  26.3   572.33
Caldwell           3        3  0.21       114     2.8  58.7     0.78
Champion          25        9  0.10        46     1.9  47.4     0.18
Coamano            2        1  0.05        77     1.9   1.9   903.82
Daphne.Major      18       11  0.34       119     8.0   8.0     1.84
```

Produce a plot of the species richness versus predictor variables. Repeat this for other predictor variables/

```
> plot(Species ~ Elevation, data = gala)
```

In the next few code blocks you'll fit a linear model and a generalised linear model to the data and look at the residuals.

```
> gala.lm1 <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
+                data = gala)
> summary(gala.lm1)

Call:
lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
    data = gala)

Residuals:
     Min       1Q   Median       3Q      Max
-111.679  -34.898   -7.862   33.460  182.584

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
Nearest      0.009144   1.054136   0.009 0.993151
Scruz       -0.240524   0.215402  -1.117 0.275208
Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared:  0.7658,        Adjusted R-squared:  0.7171
F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07

> plot(resid(gala.lm1) ~ predict(gala.lm1))
> abline(h = 0)

> gala.glm1 <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
+                  data = gala, family = poisson)
> summary(gala.glm1)

Call:
glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
    Adjacent, family = poisson, data = gala)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.2752  -4.4966  -0.9443   1.9168  10.1849

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
```

```
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

```
> layout(matrix(1:2, ncol = 2))
> plot(resid(gala.lm1) ~ predict(gala.lm1))
> abline(h = 0)
> plot(resid(gala.glm1) ~ predict(gala.glm1, type = "response"))
> abline(h = 0)
> layout(1)
```

Q and A

1. Using the outputs generated in this section, comment on the differences in fit between the linear and the Poisson model.
2. Do the predictor variables appear to be releated to the species richness of Galapágos islands? Which variables in particular?

3 Generalised additive models

We now return to *Darlingtonia* example and refit the logistic regression model used penalised regression splines (binomial GAM with logit link function)

```
> library("mgcv")                       # load mgcv
> ## fit a GAM logistic regression
> m1 <- gam(visited ~ s(leafHeight), data = wasp, family = binomial, method = "REML")
> summary(m1)

Family: binomial
Link function: logit

Formula:
visited ~ s(leafHeight)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.859      1.245  -2.296   0.0217 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                edf Ref.df Chi.sq p-value
s(leafHeight) 1.979  2.492  8.763  0.0355 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.435   Deviance explained = 47.4%
-REML = 12.375  Scale est. = 1         n = 42
```

Q and A

1. How many effective degrees of freedom are used by the smooth of `leafHeight`? What does this suggest about the shape of the fitted function?
2. Is their evidence to reject the null hypothesis that smooth of `leafHeight` is a flat function?
3. How much variation in the response is expain by the fitted model?

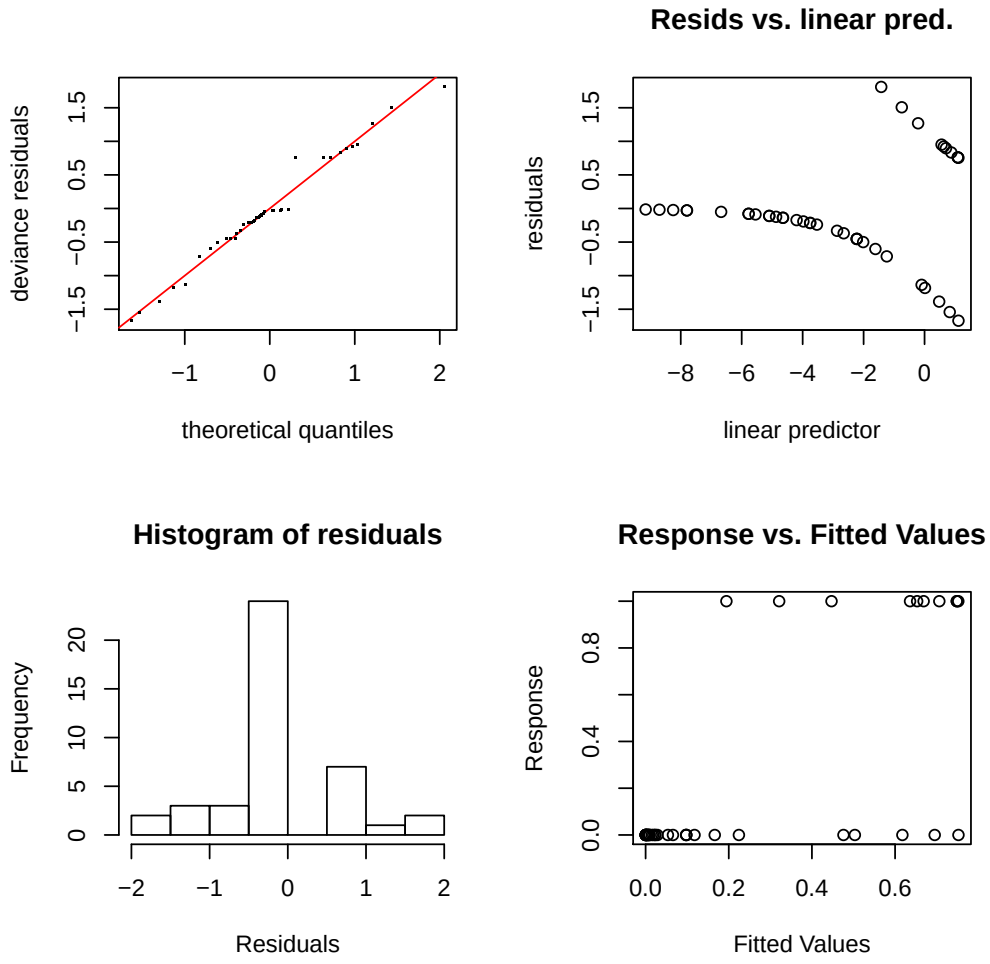
Our checks include looking at the basis dimension and producing some diagnostics plots for the fitted model. These can be done using the `gam.check()` function:

```
> gam.check(m1)

Method: REML   Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [-3.102717e-07,-3.102717e-07]
(score 12.37474 & scale 1).
Hessian positive definite, eigenvalue range [0.1985074,0.1985074].
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

                 k'  edf k-index p-value
s(leafHeight) 9.00 1.98    1.10    0.65
```



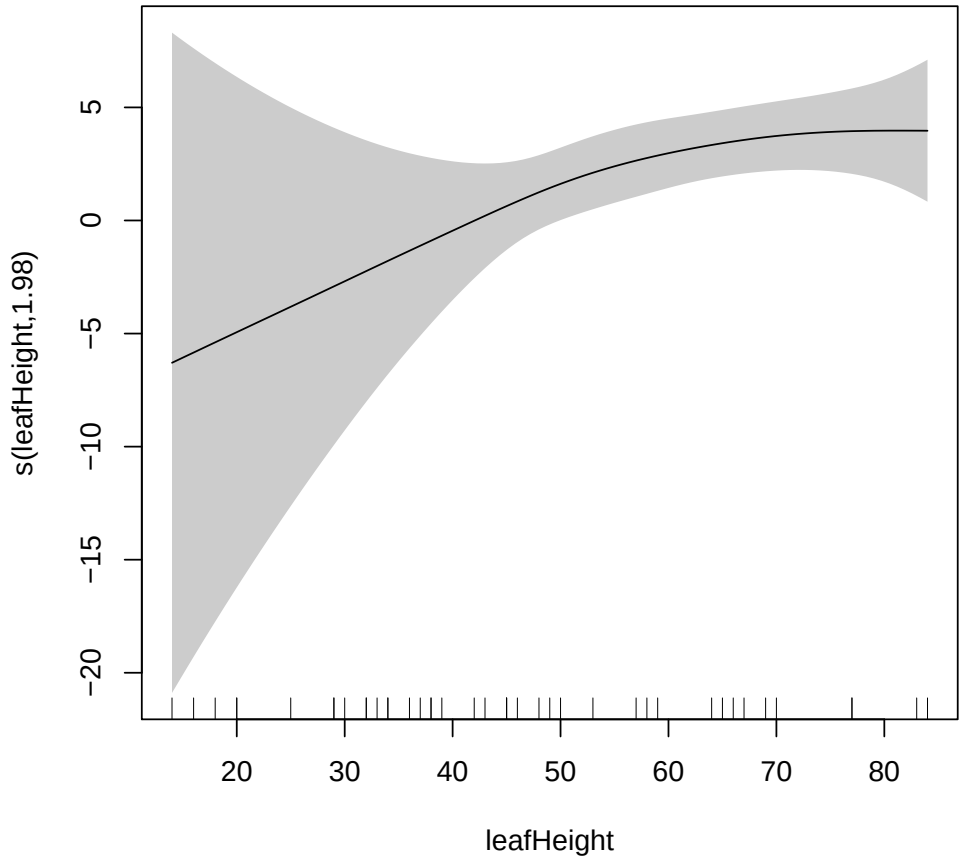
For binomial models the figures on the right, especially the top right are useless. But the ones on the left have some utility, especially the QQ-plot.

Q and A

1. Looking at the QQ-plot, do the residuals appear to follow the theortetical distribution?
2. Look at the output printed to the console; is the default basis size sufficient?

We can now plot the fitted smooth

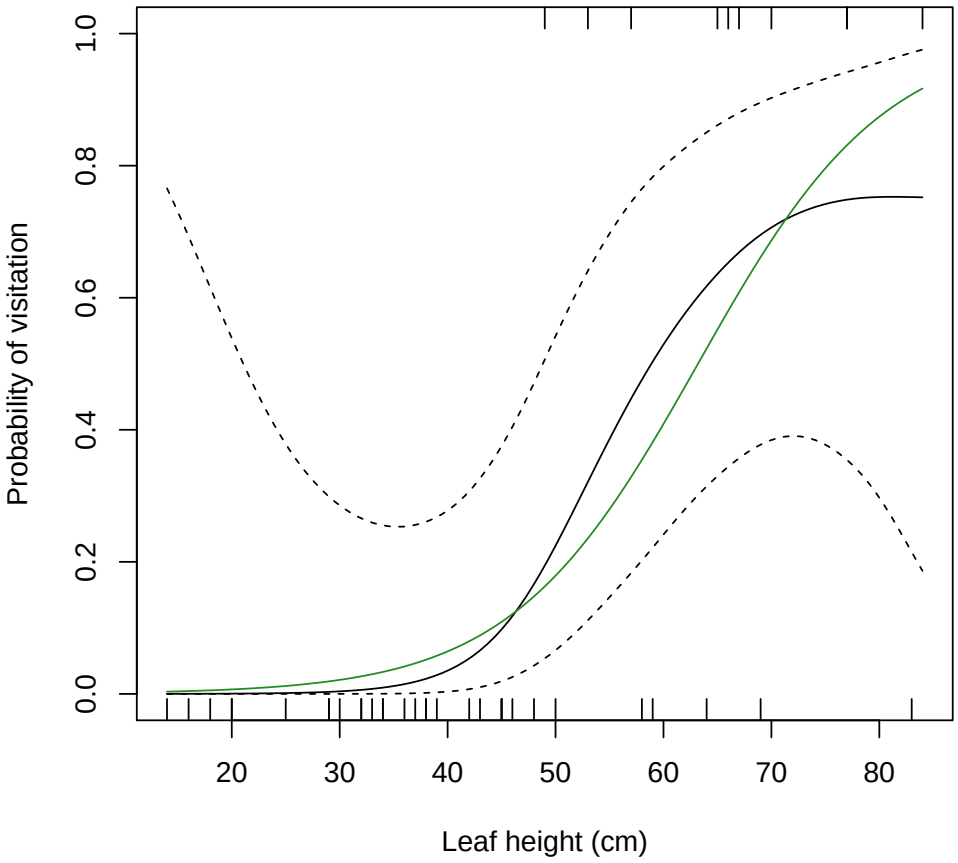
```
> plot(m1, scheme = 1, seWithMean = TRUE, unconditional = TRUE)
```



Remember, this is plot is drawn on the logit scale. We can use `predict()` as before to create a plot of the fitted model and confidence interval on the probability scale. In the last line, we add on the fitted GLM line for comparison

```
> pgam <- predict(m1, newdata = newD, type = "response")
> ## plot
> plot(pgam ~ leafHeight, data = newD, type = "l", ylim = c(0,1),
+      ylab = "Probability of visitation",
+      xlab = "Leaf height (cm)")
> with(wasp, rug(leafHeight[visited == TRUE], side = 3, lwd = 1))
> with(wasp, rug(leafHeight[visited == FALSE], side = 1, lwd = 1))
> ilogit <- family(m1)$linkinv
> pgaml <- predict(m1, newdata = newD, type = "link", se.fit = TRUE)
> crit <- 1.96
> pgam2 <- with(pgaml,
+               data.frame(fitted = ilogit(fit),
+                          upper = ilogit(fit + (crit * se.fit)),
+                          lower = ilogit(fit - (crit * se.fit)),
+                          leafHeight = newD$leafHeight))
> ## plot
```

```
> lines(upper ~ leafHeight, data = pgam2, lty = "dashed")
> lines(lower ~ leafHeight, data = pgam2, lty = "dashed")
> ## add the fitted GLM model
> lines(pred ~ leafHeight, data = newD, col = "forestgreen")
```



Do we need the smoother? To compare with the GLM, we can fit that model using **mgcv**. We do need to refit the GAM using maximum likelihood though as REML is not appropriate when you want to compare models with different fixed effects

```
> ## refit GAM using ML
> m1a <- gam(visited ~ s(leafHeight), data = wasp, family = binomial, method = "ML")
> ## fit a GLM logistic regression using mgcv
> m2 <- gam(visited ~ leafHeight, data = wasp, family = binomial, method = "ML")
> anova(m2, m1a, test = "LRT")

Analysis of Deviance Table

Model 1: visited ~ leafHeight
Model 2: visited ~ s(leafHeight)
  Resid. Df Resid. Dev         Df   Deviance  Pr(>Chi)
1        40     26.963
2        40     26.963 5.7358e-06 7.9527e-06 3.401e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In realty we probably didn't need to do that; the fact that the smoothness penalty resulted in a model with 2 degrees of freedom is strong evidence against the parametric model.