GLM2 package presentation

Maksym Lesyk, 614228

3/29/2021

GLM2 package in R

Generalized Linear Models (GLMs) are models in which response variables follow a distribution other than the normal distribution. Generalized linear models (GLMs) extend the concept of the linear regression model.
The linear model assumes that the conditional expectation of Y (which is a dependent or response variable) is equal to a linear combination of independent variables, written as:

$$Y = AX + B$$

Unfortunately, this assumption of linearity does not take into account a number of practical situations and variations, which datasets may take the form of.
For example, a continuous distribution of the error factor, as presented by the linear regression model, means that the Y variable must be continuous as well. Thus, linear regression may fail with counts (as corrected by the Poisson regression) and binary variables (e.g. logistic regression).
The term of generalized linear models goes back to the works of Nelder and Wedderburn (1972) and McCullagh and Nelder (1989).
The notation can be written down as:

$$E(Y|X) = G^{-1}(X^T\beta)$$

where G is the link function, E(Y|X) is the mean of the response variable and the expression in brackets is the standard expression denoting a certain linear combination.
The generalized linear model has two main features: 1. The distribution of the response variable (Y)
2. The link function
The distribution of Y is a member of the exponential family in this case.
We say that a distribution is a member of the exponential family if its probability mass function or density function has the following form:

$$f(y, \theta, \psi) = exp\left\{\frac{y\theta - b(\theta)}{a(\psi)}\right\} + c(y, \psi)$$

a, b and c will vary for different Y distributions. These are previously known functions. Θ is the parameter of interest and is also sometimes called the canonical parameter. Ψ is optional and is not inherent to all distributions, so it is a nuisance parameter. The two parameters are scalar.
The link function is the second element of GLMs. It relates the expected of the response to the linear predictors of the model. The function helps transform the levels of categorical variables to fit a continuous scale, which is unbound. After this transformation, the relationships between the predictors and the dependent variable can be explained using the linear regression.
In the case that the canonical parameter Θ equals the linear predictor η: If Θ= η, the link function is called the canonical link function. Some of the examples of link functions are:
Bernouli, Binomial, Poisson, Geometric, Negative Binomial, Exponential, Gamma, Normal, Inverse Gaussian.
Thus, GLMs are very useful in a way that they may help explain a wider range of data from a more natural environment.
GLMs can be fitted using either the glm package in R or its improved version, glm2 package, which is the focus of this report.
GLM2 package is the extension of the previously existing GLM package, which includes two main functions: glm2() and glm.fit2(), the former one being more commonly used with the latter one serving as a a "workhorse" ensuring that the model converges, although it can also be called upon directly.
As described by the authors themselves:
"GLM2 fits generalized linear models using the same model specification as glm in the stats package, but with a modified default fitting method that provides greater stability for models that may fail to converge using glm" While the R function glm uses step-halving to deal with certain types of convergence problems when using iteratively reweighted least squares to fit a generalized linear model, it may sometimes fail, especially when using non-standard link functions (i.e. using log link for models where independent variables are classes).
The remedy proposed by the author (Ian C. Marschner) is to impose a stricter form of step-halving than the one available in the GLM package, so that deviance is forced to decrease at every iteration.
In the stats package of R, iteratively reweighted least squares method is implemented in GLM via the glm.fit function. However, there are two specific instances of the model non-convergance:
1. The step-halving is used but it does not lead to the convergance of the model
2. The step-halving in glm.fit is never used despite the fact that the model did not converge
GLM2 is an improved version of the GLM package as it has improved convergence properties.
The main change in GLM2 is that besides the rectification of a divergent deviance of a model and testing of invalid predicted values, which was present in GLM, glm.fit2 also tests whether the deviance is lower than in the previous iteration. If it is not, then step-halving is invoked until the deviance is eventually lower.
The glm2 function is basically identical to the glm function, except fro the default fitting method, which is glm.fit2. As such, it is possible to achieve the same results with glm if glm.fit2 is specified via the method argument in glm. In this case the two functions should be identical in what they execute.
For linear models, except for the fitting method, glm2 also leads to the same results as the lm function, so the expressions look like this:

$$model = glm2(formula = x\ y, data = dat, family = \text{``}gaussian\text{''}(link = \text{``}identity))$$

which is the same as

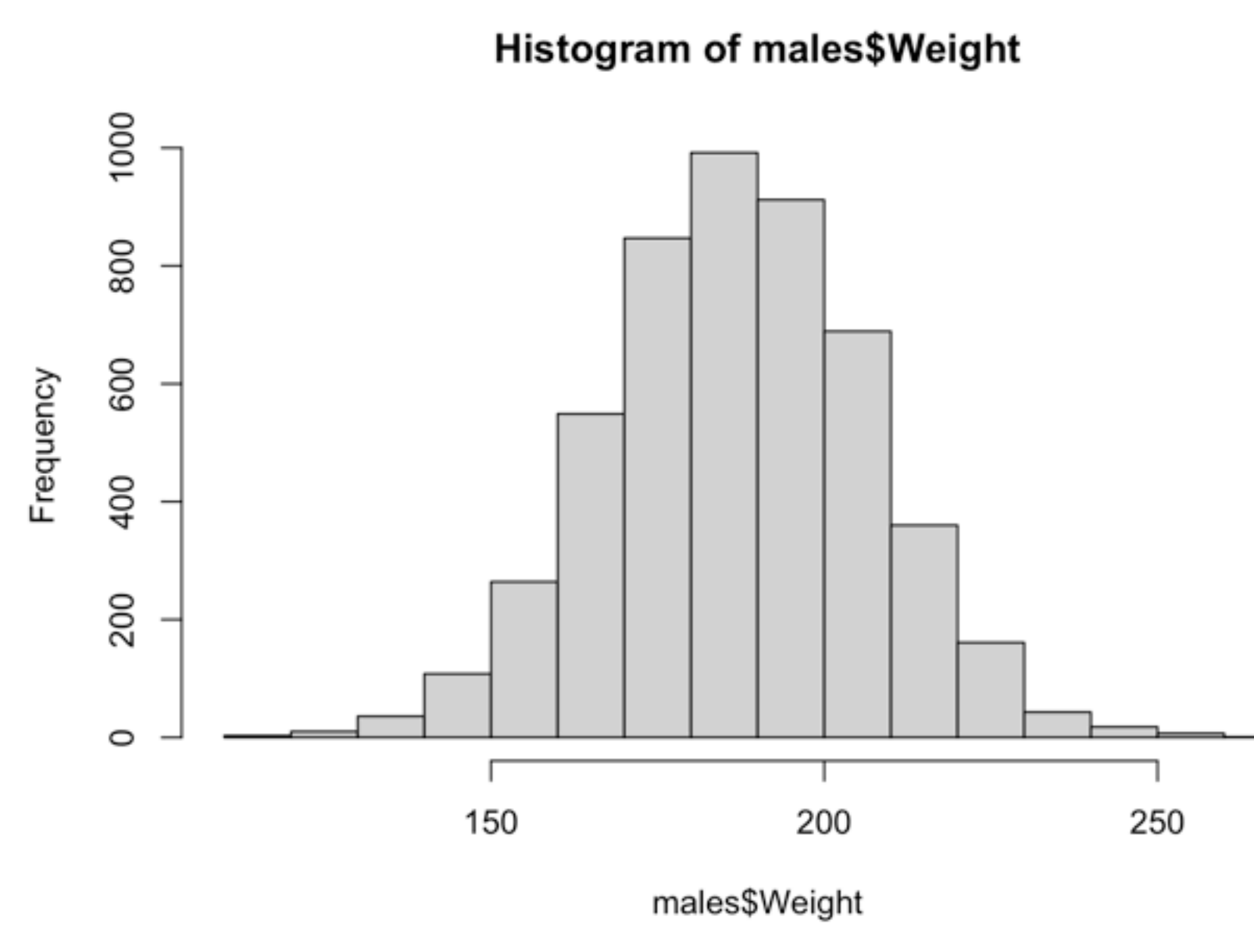
$$model = lm(formula = x\ y, data = dat)$$

glm2 can be used with default arguments where only formula, family and data have to be specified. However, it also offers an opportunity to add other arguments to the fitting process. The arguments are the following:
1. *Formula* - (which is presented as: dependent ~ independent1, independent2, independent3, etc.)
2. *Family* - (the distribution family should be specified in this argument: gaussian, poisson, binomial, quasibinomial, quasipoisson, gamma, etc.)
3. *Data* - (the dataset to be used by the glm2 function)
4. *Weights* - (it should be NULL or a numeric vector. Specifies weights of each individual entry fo the fitting process)
5. *Subset* - (specifies the subset of the main set)
6. *na.action* - (specifies what should be done when there is a NA entry. e.g. na.exclude excludes all the entries with NA from the dataset)
7. *Etastart* - (specifies starting values for the linear predictor)
8. *Mustart* - (specifies starting values for the vector of means)
9. *Offset* - (this argument can be used to specify an a priori known component to be included in the linear predictor during fitting. It should be NULL or a numeric vector of length equal to the number of cases)
10. *Control* – (this argument should specify a list of parameters for controlling the fitting process)
11. *Model* - (specifies a logical value indicating whether model frame should be included as a component of the returned value)
12. *Method* – (this argument is used to specify the method to be used in fitting the model e.g. glm.fit2)
13. *x, y* - (these are logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value)
14. *singular.ok* - (logical; if FALSE a singular fit is an error)
15. *Contrasts* - (an optional list to be used)
16. *Intercept* – (logical. It specifies whether an intercept be included in the null model)
17. *Object* – (an object inheriting from class "glm")
18. *Type* – (character, partial matching allowed. Type of weights to extract from the fitted model object. Can be abbreviated)
Each family has a certain link associated with it, however, where needed, unorthodox links may be used, in this case they are pointed out in brackets next to the family argument.
As such, see the families and respective links associated with them below:
Family - Default Link Function
Binomial - *(link="logit")*
gaussian - *(link="identity")*
Gamma - *(link="inverse")*
inverse.gaussian - *(link="1/mu^2")*
poisson - *(link="log")*
quasi - *(link="identity")*
quasibinomial - *(link="logit")*
quasipoisson - *(link="log")*
Let us see some of the examples of the glm2 package usage.
As the initial dataset we will use th data on 5000 males and 5000 females from the book by Drew Conway "Machine Learning for Hackers" with information of each individual's weight and height. Let us create a subset with only males and also draw graphs to see that our weights and heights are normally distributed.

```
library(readr)
library(ggplot2)
X01_heights_weights_genders <- read_csv("01_heights_weights_genders.csv")
```

```
##
## ── Column specification ────────────────────────────────────────────────
────────────────────────
## cols(
##   Gender = col_character(),
##   Height = col_double(),
##   Weight = col_double()
## )
```

```
males=subset(X01_heights_weights_genders, Gender == "Male")
hist(males$Weight)
```




```
hist(males$Height)
```

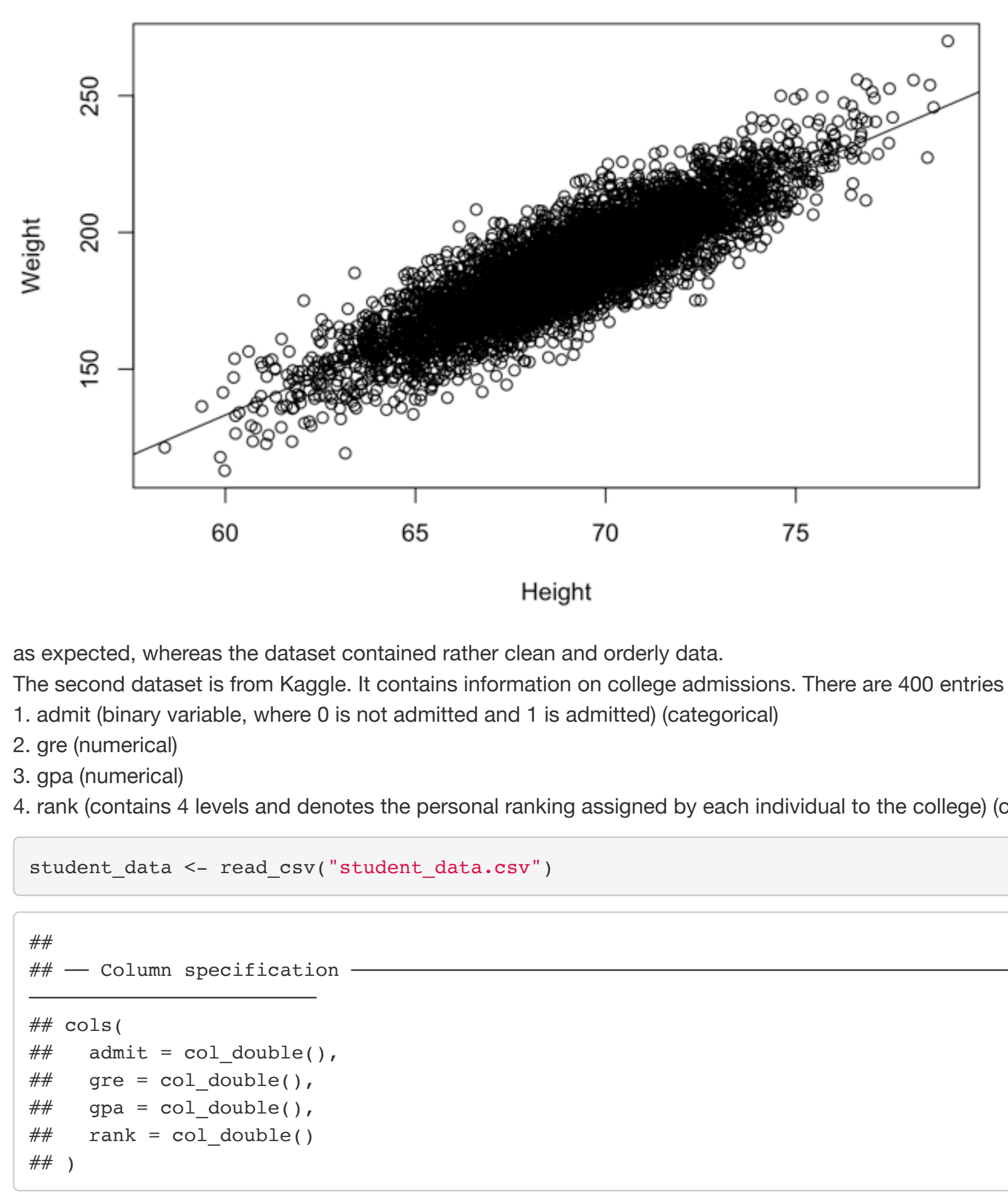
Let us then create a model using the glm2 function.

```
library(glm2)
relation=glm2(data=males,formula = Weight~Height,family=gaussian)
summary(relation)
```

```
##
## Call:
## glm2(formula = Weight ~ Height, family = gaussian, data = males)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -33.023   -6.816   -0.149    6.777   35.813
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -224.49884    3.41085  -65.82   <2e-16 ***
## Height         5.96177    0.04937  120.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 99.90465)
##
##     Null deviance: 1956079  on 4999  degrees of freedom
## Residual deviance:  499323  on 4998  degrees of freedom
## AIC: 37214
##
## Number of Fisher Scoring iterations: 2
```

The summary function helps us see some important data about our model.
As such, the uppermost part repeats the form of our function. Then, we can see information on the median and 4 quarters of our deviance residuals. Asterisks in the Coefficients section present for the graphical representation of the significance of the varibles chosen for the model. Here we can see that, naturally, Height has a significant impact on the Weight data of the selected dataset. We can also see some information on Null and Residual deviance and also the number of times it took the model to iterate in order for it to converge.
Let us join our initial data with the fitted model to see how well it fits.

```
plot(Weight ~ Height, data = males)
abline(relation)
```



As we can see the fit is rather good, as expected, whereas the dataset contained rather clean and orderly data.
The second dataset is from Kaggle. It contains information on college admissions. There are 400 entries with 4 columns:
1. admit (binary variable, where 0 is not admitted and 1 is admitted) (categorical)
2. gre (numerical)
3. gpa (numerical)
4. rank (contains 4 levels and denotes the personal ranking assigned by each individual to the college) (categorical)

```
student_data <- read_csv("student_data.csv")
```

```
##
## ── Column specification ────────────────────────────────────────────────
────────────────────────
## cols(
##   admit = col_double(),
##   gre = col_double(),
##   gpa = col_double(),
##   rank = col_double()
## )
```

```
head(student_data)
```

```
## # A tibble: 6 x 4
##   admit   gre   gpa  rank
##   <dbl> <dbl> <dbl> <dbl>
## 1     0   380  3.61     3
## 2     1   660  3.67     3
## 3     1   800  4        1
## 4     1   640  3.19     4
## 5     0   520  2.93     4
## 6     1   760  3        2
```

```
myData=student_data
myData$rank=as.factor(myData$rank)
myData$admit=as.factor(myData$admit)
```

Next, let us split the data into the test and training subsets.

```
library(caTools)
split=sample.split(myData$rank, SplitRatio = 0.85)
train=subset(myData,split=="TRUE")
test=subset(myData,split=="FALSE")
results=glm2(formula=admit~rank+gpa, family=binomial, data=train)
summary(results)
```

```
##
## Call:
## glm2(formula = admit ~ rank + gpa, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4936  -0.8927  -0.6807   1.1727   2.0341
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3967     1.1640  -2.918 0.003520 **
## rank2        -0.6558     0.3403  -1.927 0.053978 .
## rank3        -1.1960     0.3638  -3.287 0.001011 **
## rank4        -1.4791     0.4425  -3.343 0.000829 ***
## gpa           1.0287     0.3273   3.143 0.001672 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 430.96  on 339  degrees of freedom
## Residual deviance: 403.05  on 335  degrees of freedom
## AIC: 413.05
##
## Number of Fisher Scoring iterations: 4
```

4 ranks were split into 4 levels for a better representation of each rank.
As we can judge by the significance levels shown by the p-values, all of our variables are significant for the model.
Judging by the deviance, it also does not show the signs of overfitting.
It aslo took a relatively low number of times for it to converge (4).

```
res1=predict(results,train,type = "response")
```

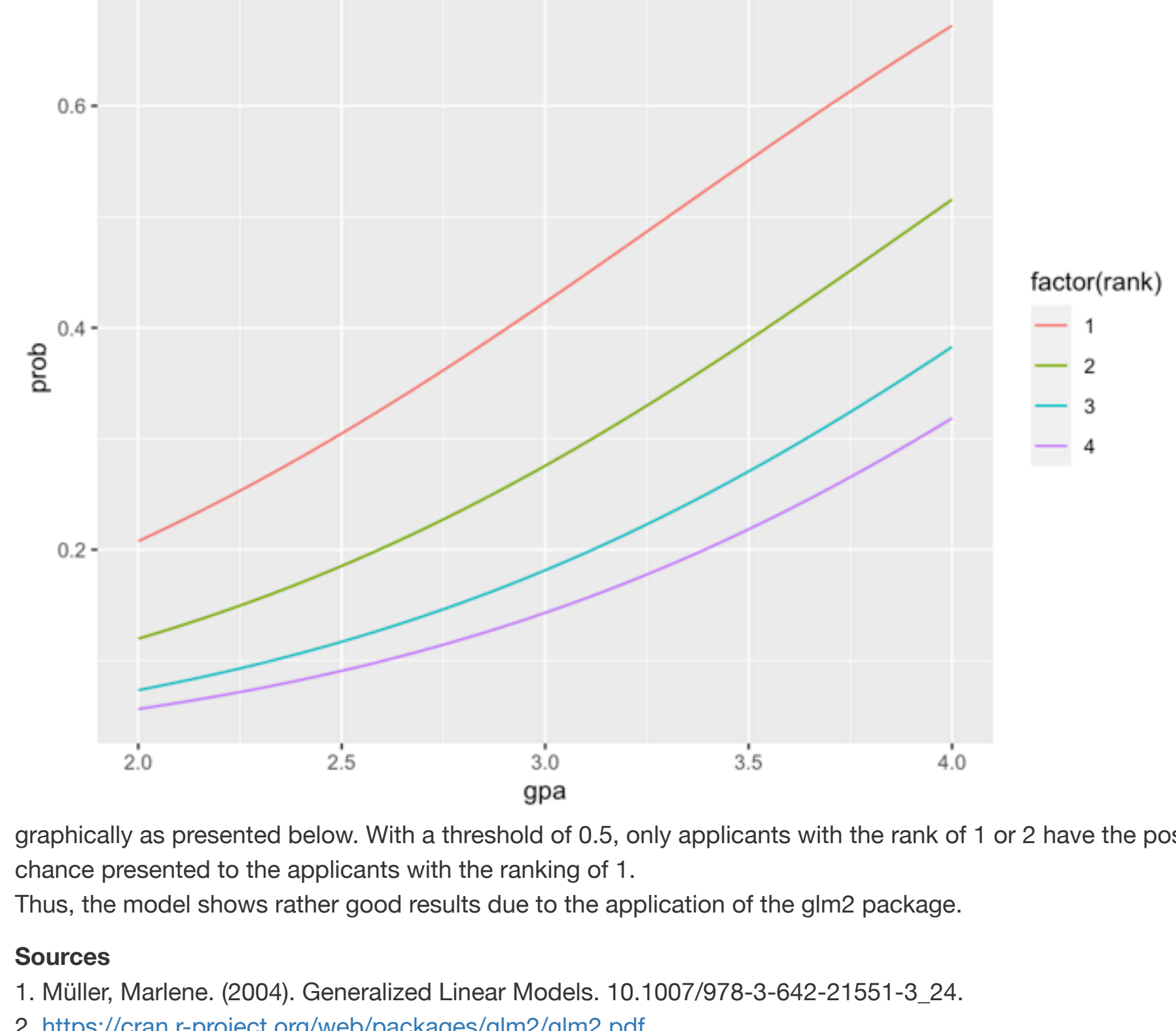
A prediction function was used to predict the outcomes based on the model obtained from the training dataset.

```
cf=table(x=train$admit, y=res1>0.5)
cf
```

```
##    y
## x   FALSE TRUE
##   0   214   14
##   1    83   29
```

A confusion matrix shows that most of the cases were predicted correctly, although the model is inclined to predict that the application was declined.

```
newdata = expand.grid(gpa=seq(2,4, 0.01), rank=1:4)
newdata$rank=as.factor(newdata$rank)
newdata$prob = predict(results, newdata, type="response")
ggplot(newdata, aes(gpa, prob, color=factor(rank), group=rank)) +
geom_line()
```



The results can also be shown graphically as presented below. With a threshold of 0.5, only applicants with the rank of 1 or 2 have the possibility to be admitted, with the largest chance presented to the applicants with the ranking of 1.
Thus, the model shows rather good results due to the application of the glm2 package.

Sources

1. Müller, Marlene. (2004). Generalized Linear Models. 10.1007/978-3-642-21551-3_24.
2. https://cran.r-project.org/web/packages/glm2/glm2.pdf
3. https://www.rdocumentation.org/packages/glm2/versions/1.2.1/topics/glm2
4.https://github.com/johnmyleswhite/ML_for_Hackers/blob/master/02-Exploration/data/01_heights_weights_genders.csv
5.https://www.kaggle.com/malapatiravi/graduate-school-admission- data/home?select=binary1.csv