7 - Multiple Logistic Regression Model

Cheng Peng

West Chester University

Contents

1 Introduction

The general multivariable linear regression model is given below.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon,$$

where y is the response variable that is assumed to be a random variable and $\epsilon \to N(0, \sigma^2)$. This also implies that

$$E[Y] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

For a binary population, the underlying random variable can only take exactly two values, say $Y = 1$ or $Y = 0$, and $P(Y = 1) = p$, then $E[Y] = 1 \times p + 0 \times (1 - p) = p$.

That is, the success probability is the expected value of the binary random variable. If we mimic the formulation of the linear regression model by setting

$$E[Y] = p = \beta_0 + \beta_1 x.$$

The issue is that the probability (left-hand side) must be between zero and one but right-hand side could be any values.

The simple logistic regression model (also called the log-odds regression model) is also formulated with the mean response $E[Y]$

$$\log \frac{E[Y]}{1 - E[Y]} = \beta_0 + \beta_1 x.$$

Let $g(t) = t/(1-t)$ (also called logit function), the simple logistic regression is re-expressed as $g(E[Y]) = \beta_0 + \beta_1 x$.

2 Multiple Logistic Regression Model

Let Y be the binary response variable and $\{x_1, x_2, \cdots, x_n\}$ be the set of predictor variables. If Y takes on either 1 or 0, the multiple logistic regression model is then defined as

$$\log \frac{E[Y]}{1 - E[Y]} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

The success probability function

$$p(x_1, x_2, \cdots, x_k) = P(Y = 1 | x_1, x_2, \cdots, x_k) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}$$

If Y takes on character values, R uses chooses the alphabetically higher value to model the above probability. For example, if Y = "disease" or "no.disease", by default, the logistic regression will be defined as

$$p(x_1, x_2, \cdots, x_k) = P(Y = "no.disease" | x_1, x_2, \cdots, x_k) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}$$

Of cause, you can also redefine the factor level of the response variable to model the probability of the desired category.

2.1 Data Requirements: Sources, Layout, and Cleansing

The logistic regression models we are discussing require an I.I.D. sample collected from a cross-sectional study design. Auto-correlation between observations is **not** allowed. For longitudinal data that involves auto-correlation, different models can be used to handle the correlation between observations taken from the same subject in the study.

The general data layout for fitting a logistic regression in R has the following form.

Table 1: Data set layout for multiple logistic regression model

Y	X1	X2		Xk
Y1	X11	X21	...	Xk1
Y2	X12	X22	...	Xk2
...	...	...	...	...

Y	X1	X2		Xk
Yn	X1n	X2n	...	Xkn

2.2 Issues of Predictor Variables and Variable Inspection-Transformation

All models have some explicit and implicit assumptions about the predictor variables and structure of the models. Unlike multiple linear regression models in which the diagnostic residual plots reveal some special patterns of potential violations of the model assumptions, in logistic regression modeling, we don't have many diagnostic tools to use. Some pre-processing procedures should be performed on predictor variables before a logistic regression model is fit to the data.

The variable inspection-transformation is an iterative process, some of the following potential issues of predictor variables may be considered in the inspection-transformation-inspection workflow.

- **Variable Types**: Predictor variables could be numeric, categorical, or a mixture of numeric and categorical.
- **Collinearity**: Predictor variables are assumed to be non-linearly correlated since the multicollinearity causes unstable estimates of the regression coefficients, hence, fails to obtain a valid model. Remedy for collinearity
  - Remove some of the highly correlated independent variables - Variable selection.
  - Perform an analysis designed for highly correlated variables such as principal components analysis or partial least squares regression - variable extraction.
  - Variable centralization.
  - Non-probabilistic variable selection - Regularization.
- **Dummy Variables**: If categorical predictor variables were numerically coded, we have to turn these numerically coded variables into factor variables. For example, the status of a disease could be "severe", "mild", and "disease-free", if a numerical coding: 2 = "severe", 1 = "mild" and 0 = "disease-free", then you need to R function **factor()** to convert the numerically coded disease status to a factor variable.
- **"Fake Variable"**: The observation ID is **NOT** a variable, you should **never** include the observation ID in any of your regression models.
- **Sparse Category Variables**: Group the categories in a meaningful way if necessary. For example, Assume that you have a data set of information about cars of different models from various manufacturers. If you want to build regression on a data set with a relatively small sample size, the use of the car-model as a categorical variable is not appropriate since too many different car models will result in too many dummy variables. From a mathematical point of view, the number of parameters should be less than the number of data points. However, from the statistical point of view, the desired sample size is 15 times the number of parameters to ensure stable estimates of model parameters.
- **Variable transformation**: - In logistic regression models, the response variable has already been transformed in the form of log odds of "success". The predictor variables could be transformed in different ways for different purposes.
  - **Association Analysis** - a transformation of predictor variables makes the interpretation of the coefficient much more difficult.
  - **Predictive Analysis** - transforming all *numerical variables* to the same scale may improve the performance of predictive models. One of the benefits of standardizing predictor variables is to make variable selection (model regularization) straightforward and interpretable.
- **Variable Discretization** - several methods can be used for the discretization: (1) **empirical approaches** include equally spaced and equal frequency, (2) **model-assisted approaches** include decision tree and k-mean. Discretization is commonly used for different purposes.

– Model interpretability and understandability - it is easier to understand continuous data (such as age) when divided and stored into meaningful categories or groups. It is commonly used in association analysis.

– Fixing the potential imbalance issues that could potentially lead to an unstable estimate of the coefficients.

2.3 Estimation and Interpretation of Regression Coefficients

As mentioned in the simple logistic regression model, regression coefficients are estimated by using the maximum likelihood approach.

Let $\{(y_1, x_{11}, x_{21}, \cdots, x_{k1}), (y_2, x_{12}, x_{22}, \cdots, x_{k2}), \cdots, (y_n, x_{1n}, x_{2n}, \cdots, x_{kn})\}$ be a random sample taken from a binary population associated with Y. x is a nonrandom predictor variable associated with Y. The logistic model is defined to be

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}.$$

The likelihood function of $(\beta_0, \beta_1, \cdots, \cdots, \beta_k)$ is given by

$$L(\beta_0, \beta_1, \cdots, \beta_k) = \prod_{i=1}^{n}\left[\frac{e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_k x_{ik}}}{1+e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_k x_{ik}}}\right]^{y_i} \times \left[\frac{1}{1+e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_k x_{ik}}}\right]^{1-y_i}$$

The maximum likelihood estimate (MLE) of $\beta_0, \beta_1, \cdots, \beta_k$, denoted by $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_k$, maximizes the above likelihood. The R build-in function **glm()** uses the MLE method to estimate parameters and reports related to MLE-based statistics.

The coefficients are interpreted similarly as used in the simple logistic regression model. To interpret β_j in the multiple logistic regression model,

$$\log\left(\frac{P[Y=1|\cdots, x_j, \cdots]}{1-P[Y=1|\cdots, x_j, \cdots]}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_{j-1}x_{j-1} + \beta_j x_j + \beta_{j+1} + \cdots + \beta_k x_k$$

If we **fix the values of all X_i except for increasing x_j by one unit**, then

$$\log\left(\frac{P[Y=1|\cdots, (x_j+1), \cdots]}{1-P[Y=1|\cdots, (x_j+1), \cdots]}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_{j-1}x_{j-1} + \beta_j (x_j+1) + \beta_{j+1}x_{j+1} + \cdots + \beta_k x_k$$

Then

$$\beta_j = \log\left(\frac{P[Y=1|\cdots, (x_j+1), \cdots]}{1-P[Y=1|\cdots, (x_j+1), \cdots]}\right) - \log\left(\frac{P[Y=1|\cdots, x_j, \cdots]}{1-P[Y=1|\cdots, x_j, \cdots]}\right) = \log\left(\frac{\frac{P[Y=1|\cdots,(x_j+1),\cdots]}{1-P[Y=1|\cdots,(x_j+1),\cdots]}}{\frac{P[Y=1|\cdots,x_j,\cdots]}{1-P[Y=1|\cdots,x_j,\cdots]}}\right)$$

Therefore, β_j (for $j = 1, 2, \cdots, k$) is the log odds ratio as explained in the simple logistic regression model.

If $\beta_j = 0$, then x_j is insignificant meaning that the odds of "success" in a subset of subjects with predictor values $\{x_1, \cdots, x_{j-1}, x_{j+1}, x_{j+1}, \cdots, x_k\}$ is equal to the odds of "success" in other subsets with predictor values $\{x_1, \cdots, x_{j-1}, x_j, x_{j+1}, \cdots, x_k\}$.

3 Building Blocks for Predictive Performance of Logistics Regression

Prediction in the logistic regression is not straightforward. The logistic regression function

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}.$$

predicts the probability of "success" but not the status of "success". In order to predict the value of Y, we still need to have a cut-off probability to define the predicted "success" and "failure". How to find the optimal cut-off probability will be addressed later module.

Next, we assume there is a cut-off probability for predicting the original value of the response variable. Most software programs as R and SAS use 0.5 as the cut-off for predicting Y.

3.1 Understanding the Performance of Medical Diagnostics

In this section, we define some performance metrics of the logistic regression as a predictive model based on the predictive error. For ease of illustration, we consider a simple model using a diagnostic test result (X = T+ or T-) to predict a disease (Y = D+ or D-) to define these performance metrics where

T+ = positive test result: numerical coding 1

T- = negative test result: numerical coding 0

D+ = diseased: numerical coding 1

D- = disease-free: numerical coding 0

The following metrics measure the predictive performance of the logistic regression model. The first two measures reflect the correct decision of the model and the last two error rates of the logistic regression model.

- **Positive Predictive Value**: P(Y = D+ | X = T+)

  $PPV = P(Y=1|X=1) = \frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}}$

- **Negative Predictive Value**: P(Y = D- | X = T-)

  $NPV = P(Y=1|X=0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$

- **False Positive Predictive Rate**: P(Y = D- | X = T+)

  $FPPV = P(Y=0|X=1) = \frac{1}{1+e^{\beta_0+\beta_1}}$

- **False Positive Predictive Rate**: P(Y = D+ | X = T-)

  $FNPV = P(Y=0|X=0) = \frac{1}{1+e^{\beta_0}}$

The above four conditional probabilities can also be estimated by calculating the corresponding relative frequencies from the following two-way contingency table - also called **confusion matrix**. For convenience, we call the above four metrics **prediction performance metrics**.

```
D1 = c("n11", "n12")
D0 = c("n21", "n22")
M=as.data.frame(rbind(D1, D0))
names(M)=c("T+", "T-")
row.names(M) = c("D+", "D-")
kable(M)
```

	T+	T-
D+	n11	n12
D-	n21	n22

The above four metrics are used by clinical diagnosis after the test was approved by the FDA since the diagnostic decision is based on the test result.

3.2 Performance Metrics Used in Clinical Trials:

Now, let's consider the case that a manufacturer conducting a clinical phase II trial and submitting the results for FDA approval. The FDA uses the following metrics in the approval process.

- **Sensitivity**: P(T+ | D+)
- **Specificity**: P(T- | D-)
- **False Negative Rate**: P(T- | D+)
- **False Positive Rate**: P(T+ | D-)

The above metrics are well-defined since the disease status of subjects is known in the clinical trial. The estimated values of these metrics can be found in the clinical data. For convenience, we call the above four metrics **Validation Performance Metrics**.

3.3 Remarks

Here are several remarks on the above two sets of performance metrics.

- The **prediction performance metrics** are dependent on the choice of the cut-off "success" probability. They can be estimated from a fitted logistic regression model.
- The **validation performance metrics** are defined based on the data with known disease status. A proposed diagnostic test is good if both sensitivity and specificity are high.
- Thinking about the logistic regression model you developed as "a diagnostic test" (since it can predict the status of a disease), which sets of metrics you should use to show the goodness of your model? The answer is the set of **validation performance metrics**.
- **Sensitivity and Specificity** are the basic building blocks used to define various performance metrics to assess the goodness of the predictive model using the **testing data** with known response values. This will be one of the major topics in the next module.

4 Case Study

In this case study, we still use the diabetes data that was used in the last module.

4.1 Data and Variable Descriptions

There are 9 variables in the data set.

1. **pregnant**: Number of times pregnant
2. **glucose**: Plasma glucose concentration (glucose tolerance test)
3. **pressure**: Diastolic blood pressure (mm Hg)
4. **triceps**: Triceps skin fold thickness (mm)
5. **insulin**: 2-Hour serum insulin (mu U/ml)

6. **mass**: Body mass index (weight in kg/(height in m)^2)
7. **pedigree**: Diabetes pedigree function
8. **age**: Age (years)
9. **diabetes**: Class variable (test for diabetes)

The data is sourced from the mlbench library in R. After removing all records with missing values, the final analytic dataset consists of 392 complete cases.

```
#library(mlbench)
data(PimaIndiansDiabetes2)           # load the data to R work-space
diabetes.0 = PimaIndiansDiabetes2    # make a copy of the data for data cleansing
diabetes = na.omit(diabetes.0)       # Delete all records with missing components
#head(diabetes)
```

4.2 Research Question

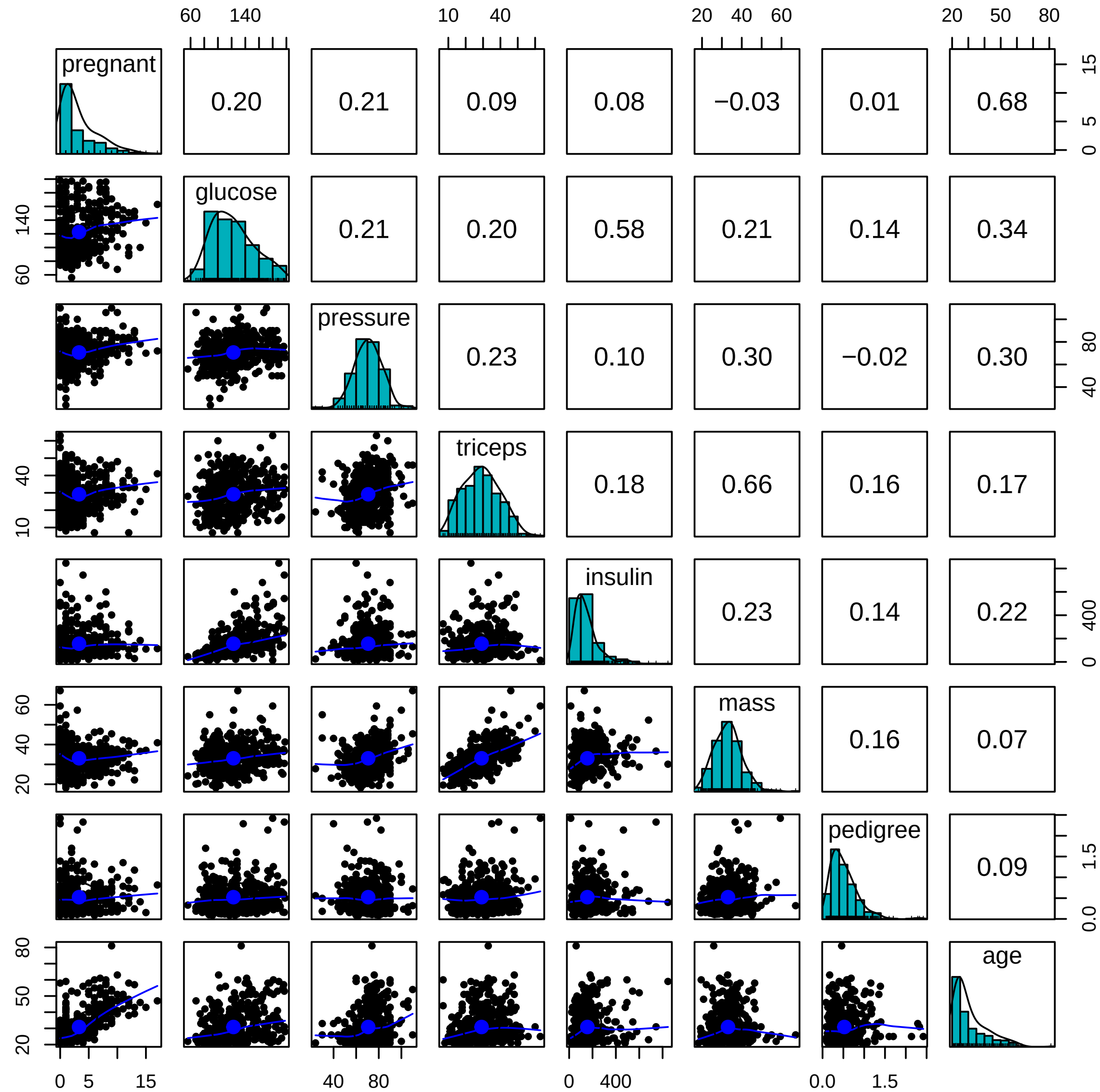
This case study seeks to identify the risk factors for diabetes. The response variable is the status of diabetes (`diabetes`: `pos = diabetes` and `neg = diabetes free`).

The response variable is a binary variable, we attempt to use the logistic regression model to address the research question.

4.3 Exploratory Analysis

To begin, we generate pairwise scatter plots for all numerical features to identify highly correlated predictors and to assess their distributional characteristics.

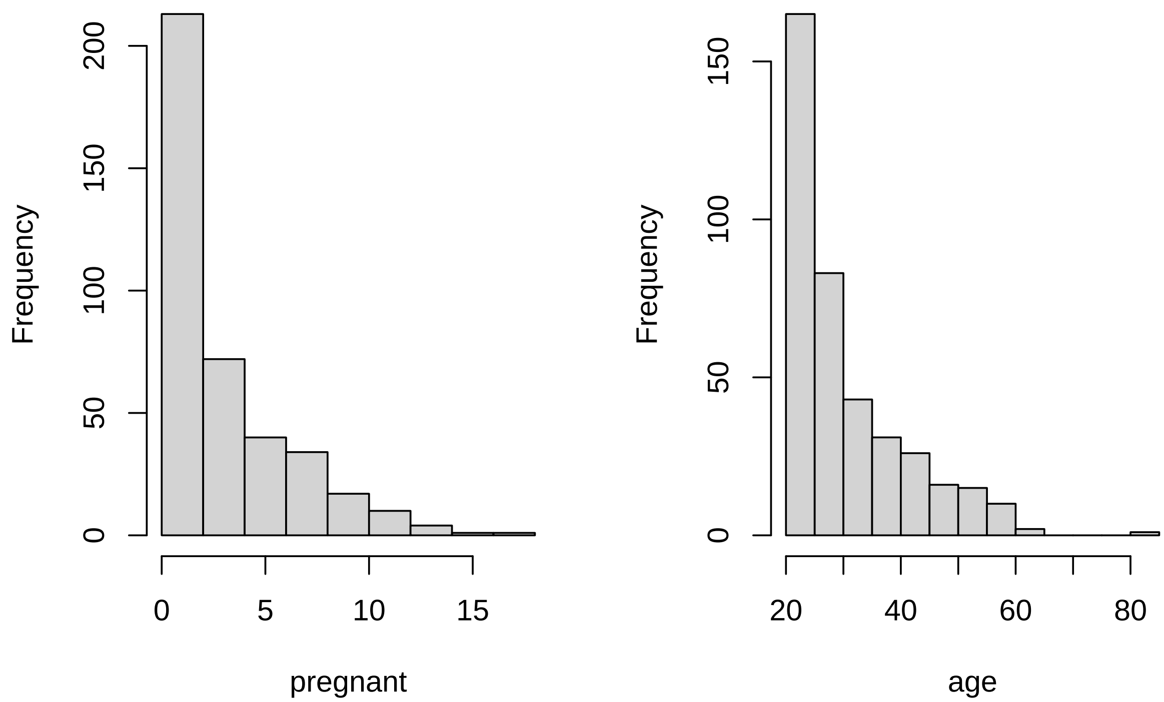
```
#library(psych)
pairs.panels(diabetes[,-9],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From the correlation matrix plot, we can see several patterns in the predictor variables.

- All predictor variables are unimodal. But **pregnant** and **age** are significantly skewed. We next take a close look at the frequency distribution of these two variables.

```
par(mfrow=c(1,2))
hist(diabetes$pregnant, xlab="pregnant", main = "")
hist(diabetes$age, xlab = "age", main = "")
```



Based on the above histogram and the best clinical practices, we discretize **pregnant** and **age** in the following.

```
preg = diabetes$pregnant
grp.preg = preg
grp.preg[preg %in% c(4:6)] = "4-6"
grp.preg[preg %in% c(7:9)] = "7-9"
grp.preg[preg %in% c(10:17)] = "10+"
##
age = diabetes$age
##
grp.age = age
grp.age[age %in% c(21:24)] = "21-25"
grp.age[age %in% c(25:30)] = "25-30"
grp.age[age %in% c(31:40)] = "31-40"
grp.age[age %in% c(41:50)] = "41-50"
grp.age[age %in% c(51:99)] = "50 +"
## added to the diabetes data set
diabetes$grp.age = grp.age
diabetes$grp.preg = grp.preg
```

- A moderate correlation is observed in several pairs of variables: **age** v.s. **pregnant**, **glucose** v.s. **insulin**, and **triceps** v.s. **mass**. We will not drop any of these variables for the moment but will perform an automatic variable selection process to remove potential redundant variables since a few of them will be forced to be included in the final model.
- Since our goal is association analysis, we will not perform variable transformations for the time being.

4.4 Building the Multiple Logistic Regression Model

Based on the exploratory analysis above, the two engineered variables will be used in the following model building. We adopt the convention of defining a scope for the logistic model and then using automatic model selection to identify the optimal model within that scope.

4.4.1 The Biggest Model

We begin by constructing a full model that includes all predictor variables potentially associated with the development of diabetes.

```
full.model = glm(diabetes ~grp.preg+glucose+pressure+triceps+insulin+mass+pedigree+grp.age,
          family = binomial(link = "logit"),  #  logit(p) = log(p/(1-p))!
          data = diabetes)
pander(summary(full.model)$coef,
      caption="Summary of inferential statistics of the full model")
```

Table 3: Summary of inferential statistics of the full model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-8.75	1.333	-6.564	5.256e-11
grp.preg1	-0.3978	0.5093	-0.781	0.4348
grp.preg10+	0.3935	0.7903	0.498	0.6185
grp.preg2	-0.2959	0.5571	-0.5311	0.5954
grp.preg3	0.3843	0.5701	0.6741	0.5003
grp.preg4-6	-0.9141	0.5668	-1.613	0.1068
grp.preg7-9	-0.2284	0.6515	-0.3507	0.7258
glucose	0.03825	0.00605	6.322	2.575e-10
pressure	-0.007294	0.0122	-0.5978	0.5499
triceps	0.01077	0.01807	0.5963	0.5509
insulin	-0.0002279	0.001385	-0.1645	0.8693
mass	0.05834	0.02849	2.047	0.04061
pedigree	1.041	0.4462	2.334	0.01961
grp.age25-30	1.083	0.4193	2.583	0.009805
grp.age31-40	1.541	0.4909	3.138	0.0017
grp.age41-50	2.26	0.6145	3.678	0.000235
grp.age50 +	2.001	0.7175	2.79	0.005277

Before creating the reduced model, we explore potential multicollinearity among the predictor variables. The following table shows the VIF for each individual variable.

```
vif(full.model)
```

```
             GVIF Df GVIF^(1/(2*Df))
grp.preg 2.937714  6        1.093958
glucose  1.423151  1        1.192959
pressure 1.240841  1        1.113931
triceps  1.698893  1        1.303416
insulin  1.434592  1        1.197745
mass     1.930566  1        1.389448
pedigree 1.056193  1        1.027713
grp.age  2.945520  4        1.144578
```

All GVIF values are less than 5. There are no significant multicollinearity issues in this data set.

4.4.2 Reduced Model

In applied settings, it is often necessary to include variables of practical importance in a final model, even if they lack statistical significance. Based on prior clinical knowledge, the variables `insulin`, `BMI`, and `pedigree` are considered essential risk factors in this diabetes study. Consequently, these three variables will be forced into the final model, defining the minimal adequate model.

```r
reduced.model = glm(diabetes ~ insulin + mass + pedigree,
          family = binomial(link = "logit"),  # logit(p) = log(p/(1-p))!
          data = diabetes)
pander(summary(reduced.model)$coef,
      caption="Summary of inferential statistics of the reduced model")
```

Table 4: Summary of inferential statistics of the reduced model ### Automatical Model Selection

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-4.329	0.6464	-6.697	2.129e-11
insulin	0.004798	0.001084	4.429	9.485e-06
mass	0.06733	0.01785	3.772	0.000162
pedigree	1.078	0.3602	2.993	0.002764

The reduced model may use insufficient information, resulting in underfitting (missing important effects), while the full model can lead to overfitting (modeling noise and failing to generalize). To address this, we use automatic variable selection to find a simpler, more interpretable, and better-performing model by intelligently choosing a subset of predictors that includes the important variables from the reduced model while avoiding the noise of the full model.

```r
## automatic variable selection
#library(MASS)
final.model.forward = stepAIC(reduced.model,
                      scope = list(lower=formula(reduced.model),upper=formula(full.model)),
                      direction = "forward",   # forward selection
                      trace = 0   # do not show the details
                      )
pander(summary(final.model.forward)$coef,
      caption="Summary of inferential statistics of the final model")
```

Table 5: Summary of inferential statistics of the final model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-9.534	1.094	-8.714	2.933e-18
insulin	-0.0005617	0.001357	-0.414	0.6789
mass	0.07031	0.02124	3.311	0.0009298
pedigree	1.057	0.4305	2.454	0.01411
glucose	0.0387	0.005881	6.58	4.692e-11
grp.age25-30	1.001	0.3918	2.555	0.01061
grp.age31-40	1.368	0.4164	3.285	0.001018
grp.age41-50	2.196	0.4712	4.661	3.152e-06
grp.age50 +	1.928	0.5803	3.322	0.000893

4.4.3 Performance Comparison

We next use deviance and AIC as measures to assess the performance of the above three models. The model with the lowest AIC value is generally preferred, as it indicates a superior balance between goodness-of-fit and model complexity, penalizing unnecessary parameters. Similarly, an analysis of deviance, which measures the discrepancy between the model and the observed data, allows us to perform likelihood-ratio tests to formally compare nested models. By examining these two criteria together, we can identify the most parsimonious model that best explains the underlying structure of the data without overfitting.

```
## Other global goodness-of-fit
global.measure=function(s.logit){
dev.resid = s.logit$deviance
dev.0.resid = s.logit$null.deviance
aic = s.logit$aic
goodness = cbind(Deviance.residual =dev.resid, Null.Deviance.Residual = dev.0.resid,
      AIC = aic)
goodness
}
goodness=rbind(full.model = global.measure(full.model),
      reduced.model=global.measure(reduced.model),
      final.model=global.measure(final.model.forward))
row.names(goodness) = c("full.model", "reduced.model", "auto.selected.model")
pander(goodness, caption ="Comparison of global goodness-of-fit statistics")
```

Table 6: Comparison of global goodness-of-fit statistics

	Deviance.residual	Null.Deviance.Residual	AIC
full.model	324.9	498.1	358.9
reduced.model	434.7	498.1	442.7
auto.selected.model	334.2	498.1	352.2

The above results shows that

- The **reduced model** has the smallest AIC, but it could be underfitted.
- The **full.model** and the **auto.selected.model** are have similar **residual deviance**.

We next use the likelihood ratio chi-squared test to see whether the the **full.model** and the **auto.selected.model** are significantly different.

The chi-square test statistic is defined to be

$$\text{Residual.Deviance}_{\text{auto.selected.model}} - \text{Residual.Deviance}_{\text{full.model}} = 334.2 - 324.9 = 9.3.$$

The degrees of freedom is

$$DF = \text{Num of variables in the full.model} - \text{Num of variables in the auto.selected.model} = 17 - 9 = 8$$

The p-value of the above likelihood ratio test is given by

$$\text{p-value} = P(\chi^2_8 > 9.3) = 0.3176239$$

Since the p-value indicates that the two models are not statistically different, we select the **auto.selected.model** as our final model in accordance with the principle of parsimony.

4.5 Final Model

In the exploratory analysis, we observed three pairs of variables are linearly correlated. After automatic variable selection, triceps and age were dropped out from the final model. Both insulin and glucose are still in the model. Although insulin is statistically insignificant, we still include it in the model since it is clinically important.

```
# Odds ratio
model.coef.stats = summary(final.model.forward)$coef
odds.ratio = exp(coef(final.model.forward))
out.stats = cbind(model.coef.stats, odds.ratio = odds.ratio)
kable(out.stats,caption = "Summary Stats with Odds Ratios")
```

Table 7: Summary Stats with Odds Ratios

	Estimate	Std. Error	z value	Pr(>|z|)	odds.ratio
(Intercept)	-9.5340546	1.0941037	-8.7140321	0.0000000	0.0000723
insulin	-0.0005617	0.0013568	-0.4140043	0.6788710	0.9994384
mass	0.0703082	0.0212351	3.3109430	0.0009298	1.0728388
pedigree	1.0567094	0.4305469	2.4543423	0.0141143	2.8768887
glucose	0.0387018	0.0058814	6.5804090	0.0000000	1.0394605
grp.age25-30	1.0012238	0.3918271	2.5552695	0.0106106	2.7216105
grp.age31-40	1.3679308	0.4163693	3.2853782	0.0010185	3.9272159
grp.age41-50	2.1962925	0.4712387	4.6606796	0.0000032	8.9916151
grp.age50 +	1.9277491	0.5802575	3.3222305	0.0008930	6.8740202

The interpretation of the odds ratios is similar to the case of simple logistic regression. The group-age variable **grp.age** has five categories. The baseline category is aged 21-24. We can see from the above table inferential table that the odds of getting diabetes increase as age increases. For example, the odds ratio associated with the age group 31-39 is 3.927 meaning that, given the same level of insulin, BMI, pedigree, and glucose, the odds of being diabetic in the age group of 31-40 is almost 4 times of that in the baseline group aged 21-24. But the same ratio becomes nine times when comparing the age group 41-50 with the baseline group of age 21-24.

4.6 Summary and Conclusion

The case study focused on the association analysis between a set of potential risk factors for diabetes. The initial data set has 8 numerical and categorical variables.

After exploratory analysis, we decide to re-group two sparse discrete variables **pregnant** and **age**, and then define dummy variables for the associated variables. These new group variables were used in the model search process.

Since **insulin**, **BMI**, and **pedigree** are considered to be major contributors to the development of diabetes, we include three risk factors in the final model regardless of the statistical significance.

After automatic variable selection, we obtain the final model with 4 factors, BMI, pedigree, glucose, age (with 4 dummy variables), and insulin (that is not statistically significant but clinically important).

Diabetes prediction or classification is another important practical issue. We will address this practical question in the next module.