Exercise 22
Applied Biostatistics II

Hector Arribas Arias

Julian Niklaus

Matthias Rubin

Setup of the exercise

The data set **baby.dat** contains data from a study in which clinicians measured several clinical variables of premature babies; **the response variable: Survival**, notes whether the babies survived (**Survival = 1**) or not (**Survival = 0**).

a. Fit a logistic regression model to the data, using all explanatory variables. Indicate the misclassification rate on the data set.

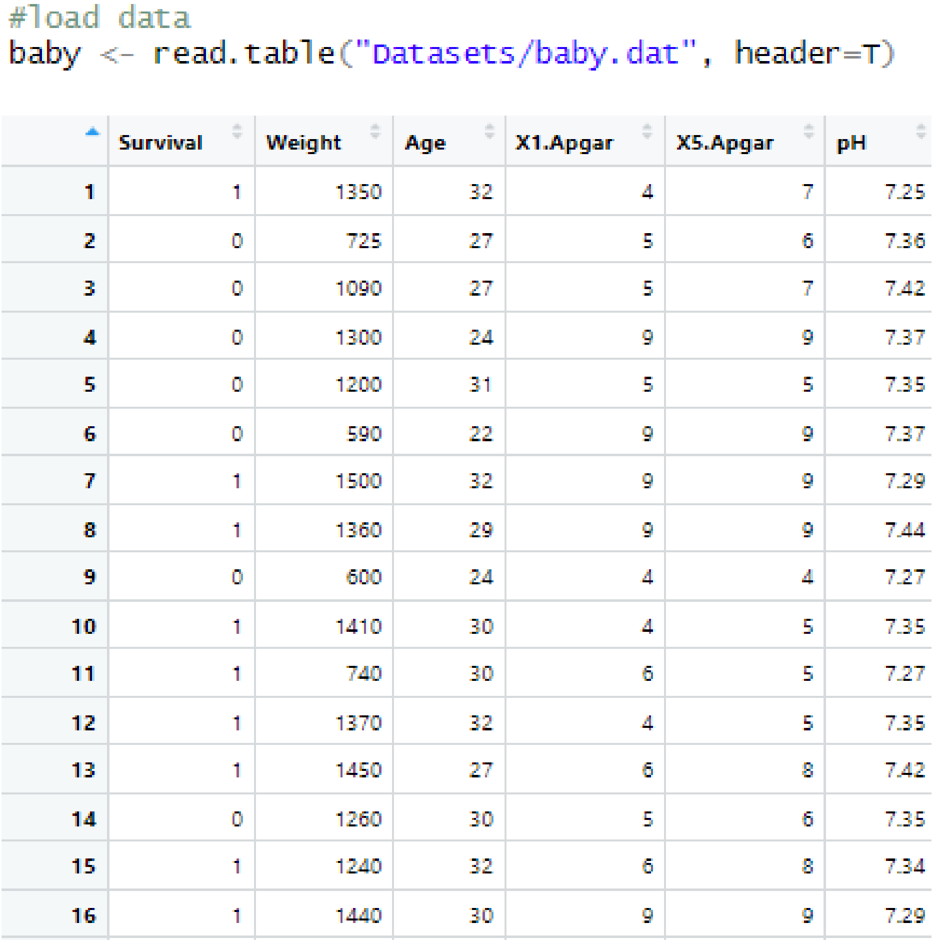
b. Start with the full model you got in a) and eliminate variables using backward selection.

c. Estimate the expected misclassification rate for the model you get in b) using leave-one-out cross validation.

d. Comment on the different misclassification rates you get.

Overview of the data

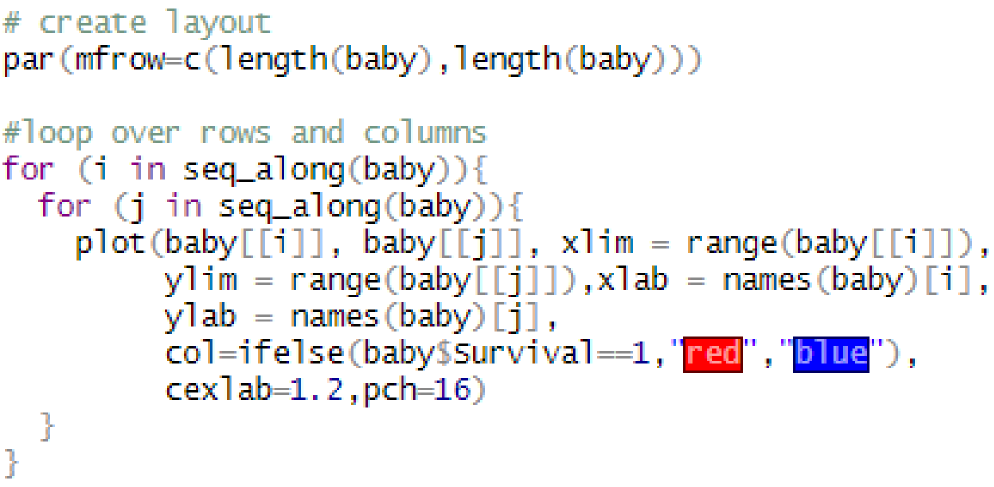
- Data from 247 premature babies
- Response variable: Survival
- 5 explanatory variables:
  - Weight in grams
  - Age in weeks after procreation
  - X1.Apgar: APGAR-Score after 1min (**A**ppearance, **P**ulse, **G**rimace, **A**ctivity, **R**espiration)
  - X5.Apgar: APGAR-Score after 5min
  - pH: measured in blood



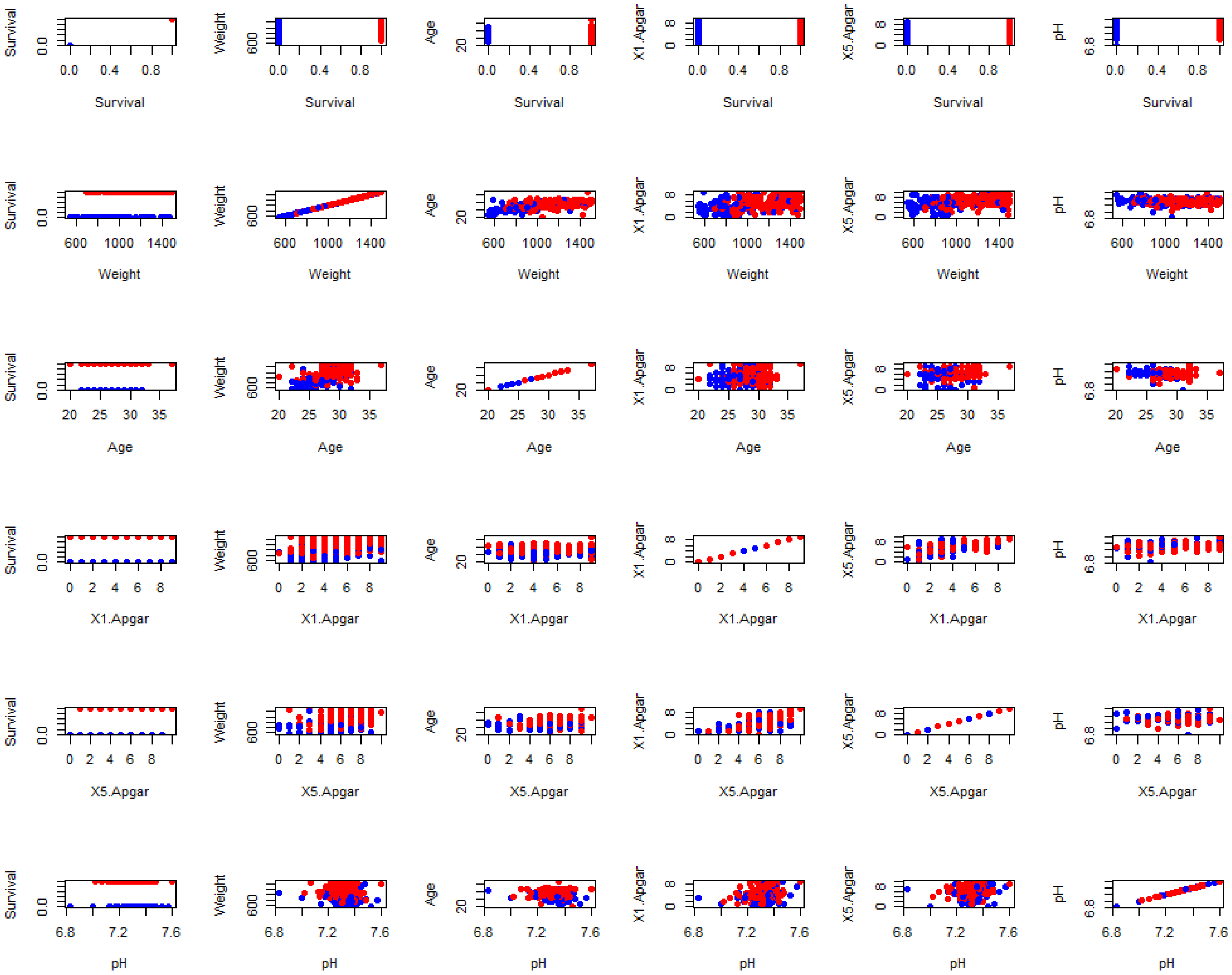
```
#load data
baby <- read.table("Datasets/baby.dat", header=T)
```

	Survival	Weight	Age	X1.Apgar	X5.Apgar	pH
1	1	1350	32	4	7	7.25
2	0	725	27	5	6	7.36
3	0	1090	27	5	7	7.42
4	0	1300	24	9	9	7.37
5	0	1200	31	5	5	7.35
6	0	590	22	9	9	7.37
7	1	1500	32	9	9	7.29
8	1	1360	29	9	9	7.44
9	0	600	24	4	4	7.27
10	1	1410	30	4	5	7.35
11	1	740	30	6	5	7.27
12	1	1370	32	4	5	7.35
13	1	1450	27	6	8	7.42
14	0	1260	30	5	6	7.35
15	1	1240	32	6	8	7.34
16	1	1440	30	9	9	7.29

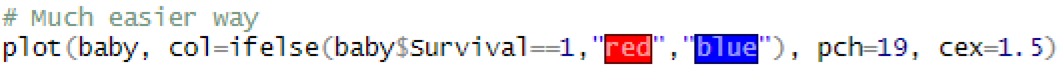
Visualization of the data



```
# create layout
par(mfrow=c(length(baby),length(baby)))

#loop over rows and columns
for (i in seq_along(baby)){
  for (j in seq_along(baby)){
    plot(baby[[i]], baby[[j]], xlim = range(baby[[i]]),
         ylim = range(baby[[j]]),xlab = names(baby)[i],
         ylab = names(baby)[j],
         col=ifelse(baby$Survival==1,"red","blue"),
         cexlab=1.2,pch=16)
  }
}
```



Visualization of the data

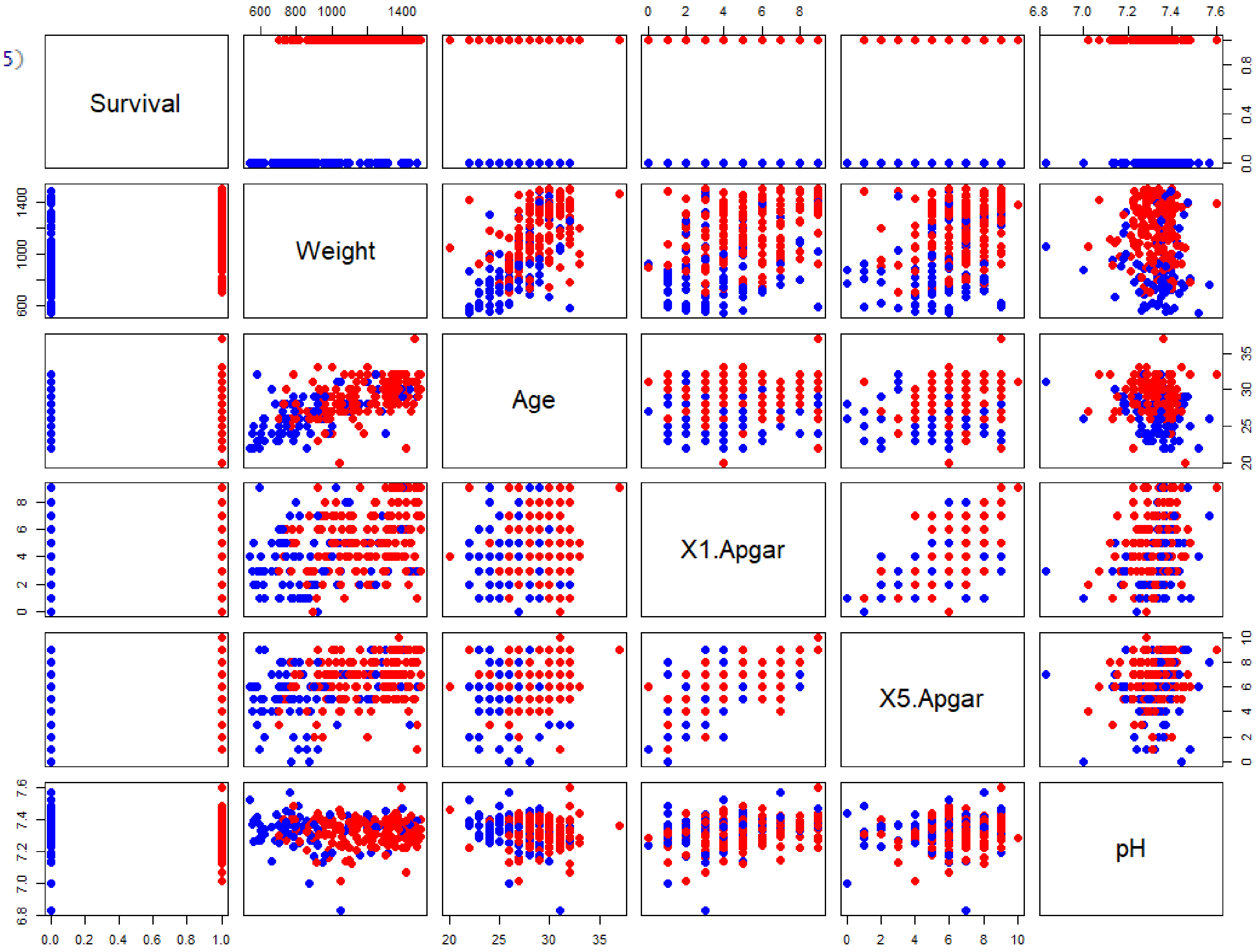


```
# Much easier way
plot(baby, col=ifelse(baby$Survival==1,"red","blue"), pch=19, cex=1.5)
```

The plots indicate the correlation between the variables, the diagonal plots are not shown as the same variable plotted against each other does not give any information.

For example weight against age you can see that babies with lower weight and which are younger (premature) die more than bigger and more mature babies

Significance is not shown

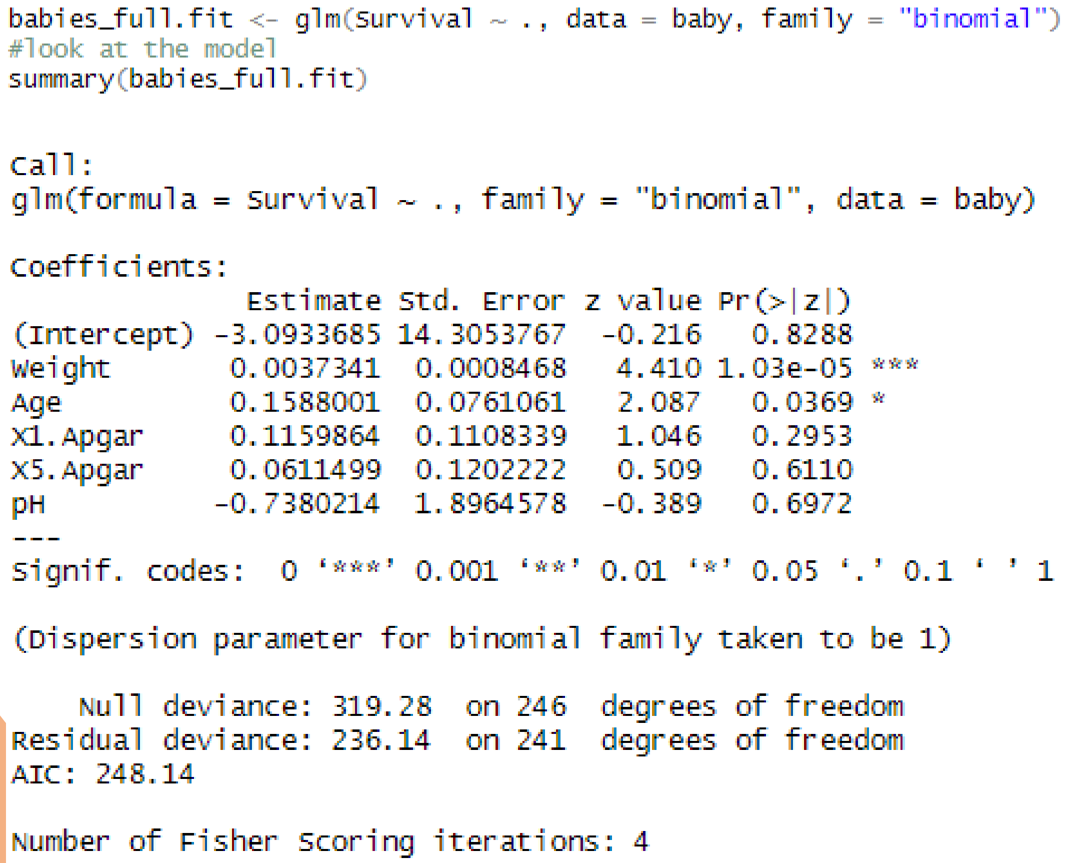


Logistic Regression model - Refresher

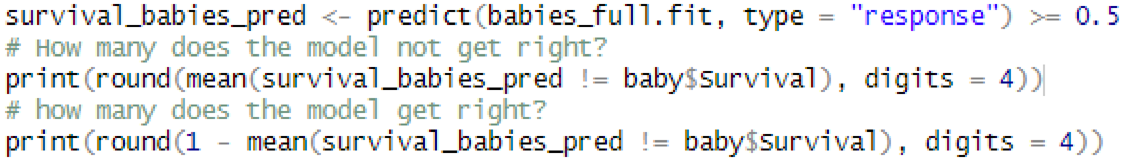
- **What is it?**
- Statistical modeling technique.
- Logistic regression uses the logistic function (also known as the sigmoid function) to model the relationship between predictor variables and the probability of a binary outcome
- Aim: Fit the posterior class probabilities by applying a logistic tranformation to pi1 (probability of a datapoint of being in a class) $\log\left(\frac{\pi_1(x_1,\ldots,x_p)}{1-\pi_1(x_1,\ldots,x_p)}\right)$

- **When is it used?**
- Binary outcome: Used when the dependent variable Y is binary (0/1 or categorical with two levels).
  The goal is to model the probability that Y=1 given predictor variables X.
- The model estimates probabilities, providing insights into the likelihood of each outcome.
- It helps in understanding the relationship between the independent variables (features) and the binary outcome.
- it can handle non-linear relationships between the features and the outcome.

a) Fit a logistic regression model to the data, using all explanatory variables. Indicate the misclassification rate on the data set.



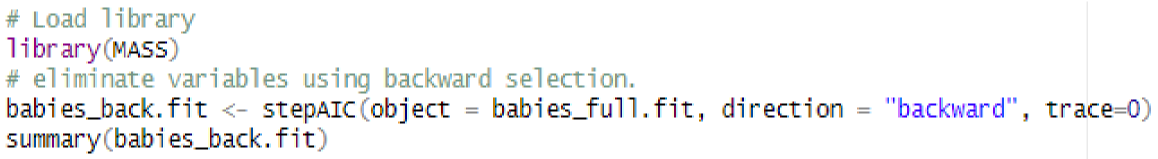
```
babies_full.fit <- glm(Survival ~ ., data = baby, family = "binomial")
#look at the model
summary(babies_full.fit)


Call:
glm(formula = Survival ~ ., family = "binomial", data = baby)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -3.0933685 14.3053767  -0.216   0.8288    
Weight       0.0037341  0.0008468   4.410 1.03e-05 ***
Age          0.1588001  0.0761061   2.087   0.0369 *  
X1.Apgar     0.1159864  0.1108339   1.046   0.2953    
X5.Apgar     0.0611499  0.1202222   0.509   0.6110    
pH          -0.7380214  1.8964578  -0.389   0.6972    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 319.28  on 246  degrees of freedom
Residual deviance: 236.14  on 241  degrees of freedom
AIC: 248.14

Number of Fisher Scoring iterations: 4
```



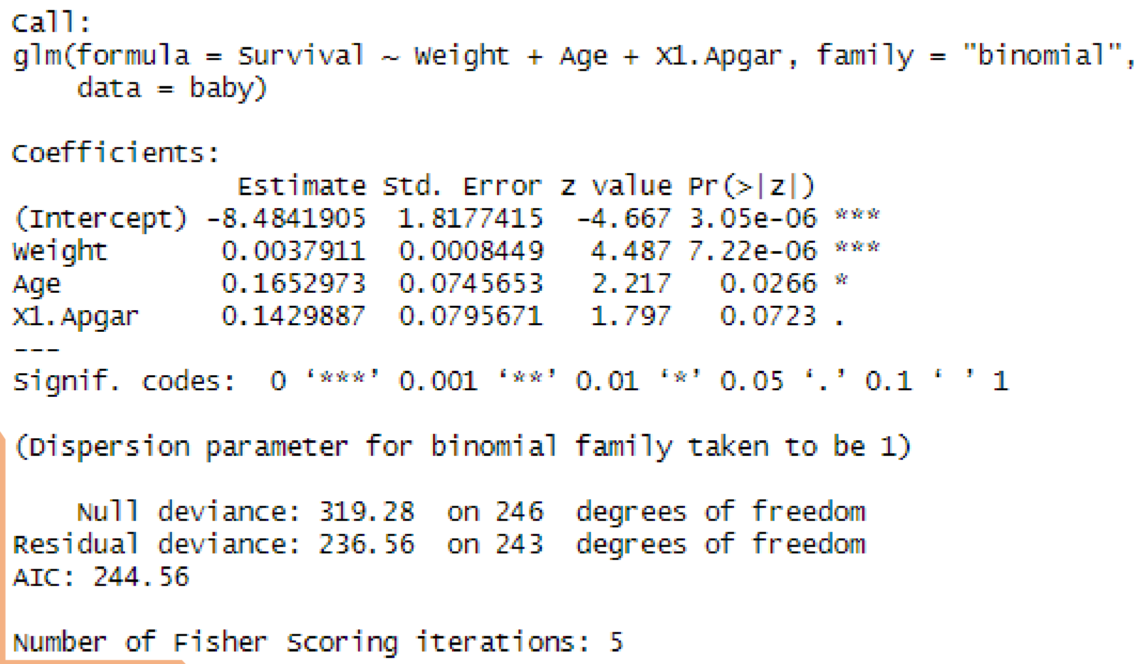
```
survival_babies_pred <- predict(babies_full.fit, type = "response") >= 0.5
# How many does the model not get right?
print(round(mean(survival_babies_pred != baby$Survival), digits = 4))
# how many does the model get right?
print(round(1 - mean(survival_babies_pred != baby$Survival), digits = 4))
```

- Misclassification rate in the full model:
    - 0.2105
- Model classifies correctly in 78.95% of the cases

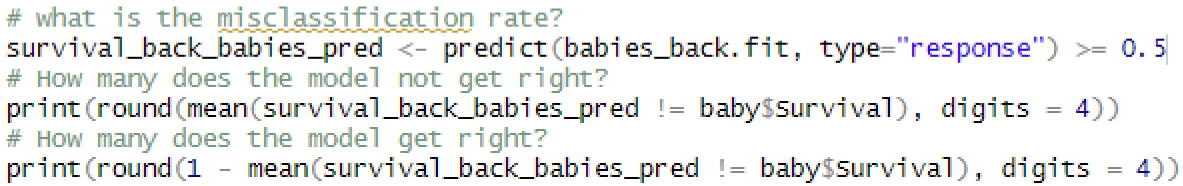
b) Start with the full model you got in a) and eliminate variables using backward selection.
Which model do you end up with? What's its misclassification rate?



```
# Load library
library(MASS)
# eliminate variables using backward selection.
babies_back.fit <- stepAIC(object = babies_full.fit, direction = "backward", trace=0)
summary(babies_back.fit)
```



```
Call:
glm(formula = Survival ~ Weight + Age + X1.Apgar, family = "binomial",
    data = baby)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -8.4841905  1.8177415  -4.667 3.05e-06 ***
Weight       0.0037911  0.0008449   4.487 7.22e-06 ***
Age          0.1652973  0.0745653   2.217   0.0266 *  
X1.Apgar     0.1429887  0.0795671   1.797   0.0723 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 319.28  on 246  degrees of freedom
Residual deviance: 236.56  on 243  degrees of freedom
AIC: 244.56

Number of Fisher Scoring iterations: 5
```



```
# what is the misclassification rate?
survival_back_babies_pred <- predict(babies_back.fit, type="response") >= 0.5
# How many does the model not get right?
print(round(mean(survival_back_babies_pred != baby$Survival), digits = 4))
# How many does the model get right?
print(round(1 - mean(survival_back_babies_pred != baby$Survival), digits = 4))
```

- AIC in full vs selected model: 248.14 vs 244.56
- Misclassification rate in the reduced model:
  - 0.2106
- Model classifies correctly in 78.54% of the cases

c) Estimate the expected misclassification rate (for babies that were not in the study) for the model you get in b) using leave-one-out cross validation.

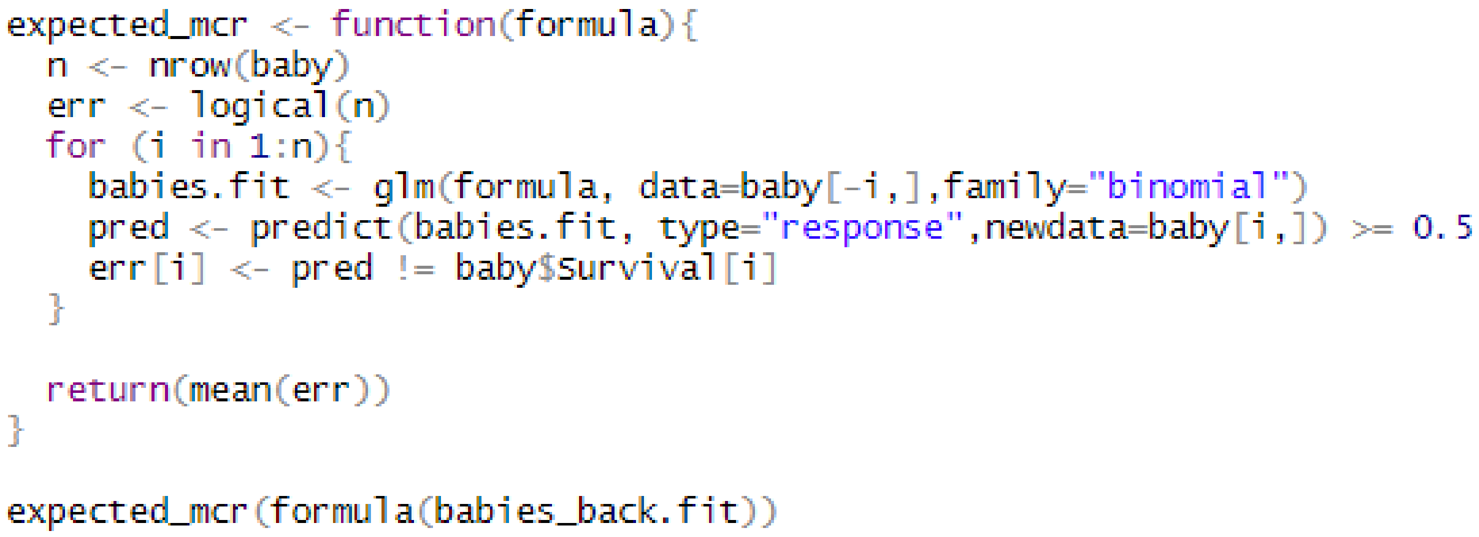
- In contrast to exercise 18, our error here is not calculated as the square between predicted and measured response variable (MSE), but the 0-1 classification error.

- **Error Calculation**: During each iteration, the model predicts whether a baby survived or not based on the features (predictor variables) using logistic regression. Then, the predicted classification is compared with the true survival status.

- **Misclassification**: If the predicted classification matches the true survival status, the error for that observation is 0 (correctly classified). If the predicted classification does not match the true survival status, the error for that observation is 1 (misclassified).

- **Mean Error Calculation**: After iterating through all observations and calculating the errors for each, the mean of these errors is taken.

c) Estimate the expected misclassification rate (for babies that were not in the study) for the model you get in b) using leave-one-out cross validation.



```r
expected_mcr <- function(formula){
  n <- nrow(baby)
  err <- logical(n)
  for (i in 1:n){
    babies.fit <- glm(formula, data=baby[-i,],family="binomial")
    pred <- predict(babies.fit, type="response",newdata=baby[i,]) >= 0.5
    err[i] <- pred != baby$Survival[i]
  }

  return(mean(err))
}

expected_mcr(formula(babies_back.fit))
```

- Expected misclassifcation rate in the reduced model:
  - [1] 0.2186235
- Model classifies correctly in 88% of the cases for **new data**

d) Comment on the different misclassification rates (or expected misclassification rates) you get in tasks a) to c).

- We can see that the misclassification rate is slightly rising from part a) to part c), although the rates are all close together: **0.2105263 -> 0.2145749 -> 0.2186235**
- **The actual misclassification rate and the expected misclassification rate of the reduced model lie very close together.**

- Where could differences come from:
  - Difference from full model to reduced model: larger, more complex models tend to overfit the data.
  - Difference from b) to c): the actual misclassification rate tends to be over-optimistic compared to the expected misclassification rate estimated by cross validation.

Additional slide: Cross validation

- **Aim:** we want to estimate the prediction performance of a model on new data

- **How:** split dataset into multiple parts, most of the data is used for training, the rest for testing the model

- => We know the true outcome of test data, so we can compare it to the prediction and make conclusions about the validity of the model

Additional slide: Leave-one-out cross validation

- **Given**: response vector Y with n measurements, design or model matrix X

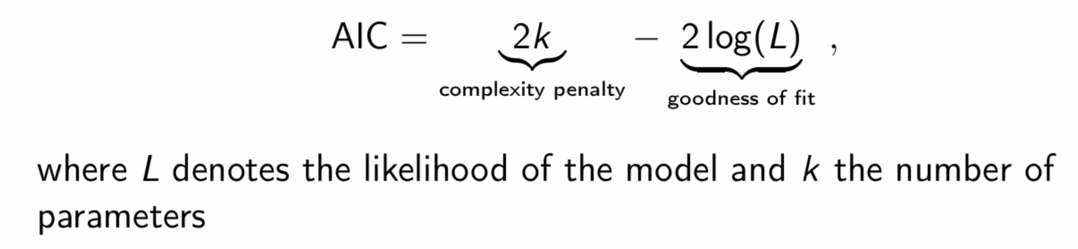
- **For each datapoint (i=1, ….,n):**
  - We remove the data point from Y and X
  - We fit a linear model on the remaining datapoints using regularized regression
  - We predict the left-out value $\hat{Y}_i$ from the model

- We calculate the mean squared error, which we use to compare the goodness of the models:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

Additional Slide: AIC backwards selection

- **AIC:** Akaike information criterion
- **Basic Idea:** Use likelihood of the model BUT penalize for the complexity of the model



$$\mathrm{AIC} = \underbrace{2k}_{\text{complexity penalty}} - \underbrace{2\log(L)}_{\text{goodness of fit}},$$

where L denotes the likelihood of the model and k the number of parameters

- **Backwards selection means:**
  - We start with the full model
  - We remove parameters stepwise, try to minimize AIC