Assignment 3

Assignment 3. Logistic regression and basis function expansion

The data file **pima-indians-diabetes.csv** contains information about the onset of diabetes within 5 years in Pima Indians given medical details. The variables are (in the same order as in the dataset):

1. Number of times pregnant.

2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test.

3. Diastolic blood pressure (mm Hg).

4. Triceps skinfold thickness (mm).

5. 2-Hour serum insulin (mu U/ml).

6. Body mass index (weight in kg/(height in m)^2).

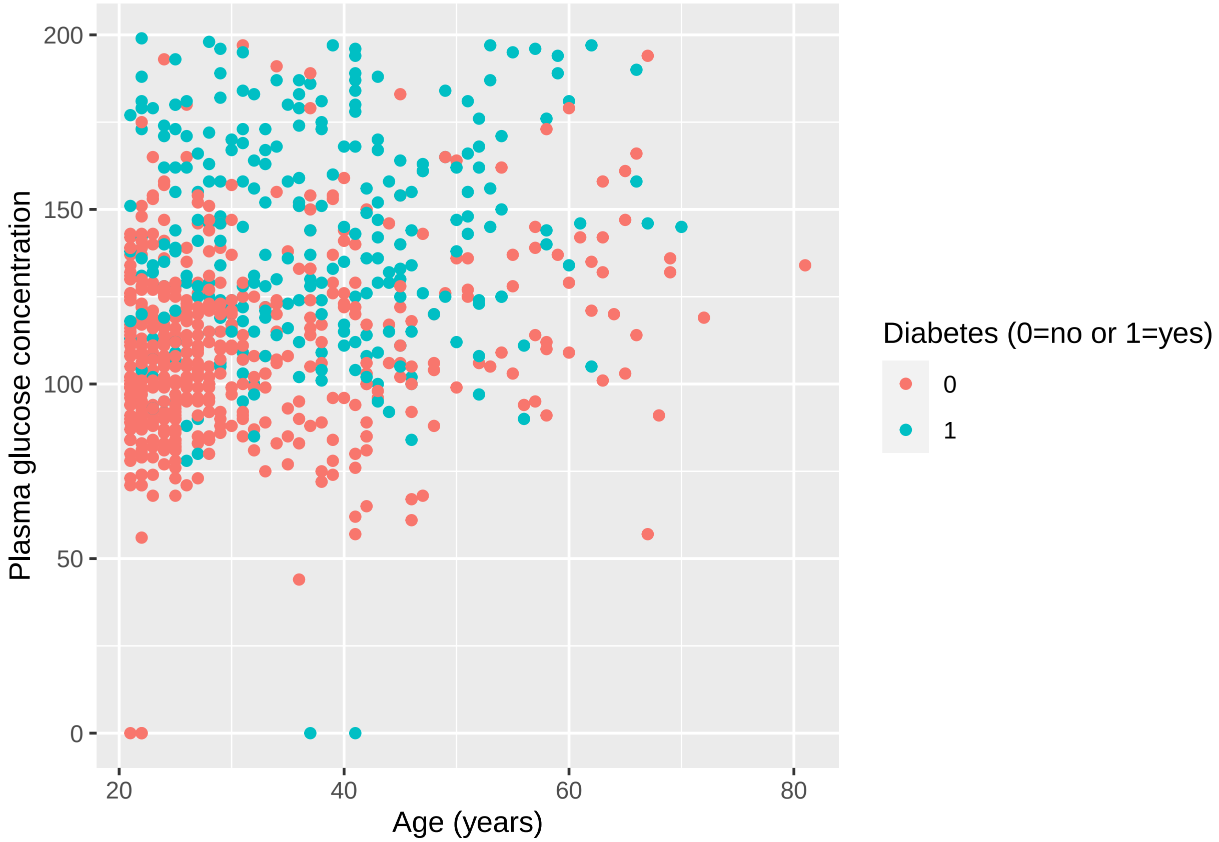
7. Diabetes pedigree function.

8. Age (years).

9. Diabetes (0=no or 1=yes)

1.Make a scatterplot showing a Plasma glucose concentration on Age where observations are colored by Diabetes levels. Do you think that Diabetes is easy to classify by a standard logistic regression model that uses these two variables as features? Motivate your answer.

```r
data_PIdiabetes <- read.csv("pima-indians-diabetes.csv", col.names = c(1:9))

data_PIdiabetes$X9 <- factor(data_PIdiabetes$X9)

ggplot(data = data_PIdiabetes, aes(x = X8, y = X2, colour = X9)) +
  geom_point() +
  labs(x = "Age (years)", y = "Plasma glucose concentration",
       colour  = "Diabetes (0=no or 1=yes)" )
```



- Yes! Motivation: Looking at the plot, one can see that the plasma glucose concentration is more higher for the group with Diabetes level 1 compared to those with level 0. This is more evident when plasma glucose concentration is 150 or higher. At such one can conclude that it will be easy to classify Diabetes by a standard logistic regression

2.Train a logistic regression model with y = Diabetes as target x_1 = Plasma glucose concentration and x_2 = Age as features and make a prediction for all observations by using $r = 0.5$ as the classification threshold. Report the probabilistic equation of the estimated model (i.e., how the target depends on the features and the estimated model parameters probabilistically). Compute also the training misclassification error and make a scatter plot of the same kind as in step 1 but showing the predicted values of Diabetes as a color instead. Comment on the quality of the classification by using these results

- Logistic regression analysis belongs to the class of generalized linear models. In R generalized linear models are handled by the glm() function. The function is written as glm(response ~ predictor, family = binomial(link = "logit"), data). Please note that logit is the default for binomial; thus, we do not have to type it explicitly. The glm() function returns a model object, therefore we may apply extractor functions, such as summary(), fitted() or predict, among others, on it. However, please note that the output numbers are on the logit scale. To actually predict probabilities we need to provide the predict() function an additional argument type = "response"

```r
# X2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
# X8. Age (years).
# X9. Diabetes (0=no or 1=yes).

glm_diabete <- glm(X9 ~ X2 + X8, data_PIdiabetes, family = binomial)

fit <- ifelse(fitted(glm_diabete) >= 0.5, 1, 0)
```

```
glm_diabete$coefficients
```

```
## (Intercept)          X2          X8 
## -5.89785793  0.03558250  0.02450157 
```

- we have

  $$g(x) = \frac{1}{1 + e^{-\theta^T x}}$$

  and use coefficients we got:

  $$g(x) = \frac{1}{1 + e^{-(-5.89785793 + 0.03558250 x_1 + 0.02450157 x_2)}}$$

  while $x_1 =$ Plasma glucose concentration and $x_2 =$ age.

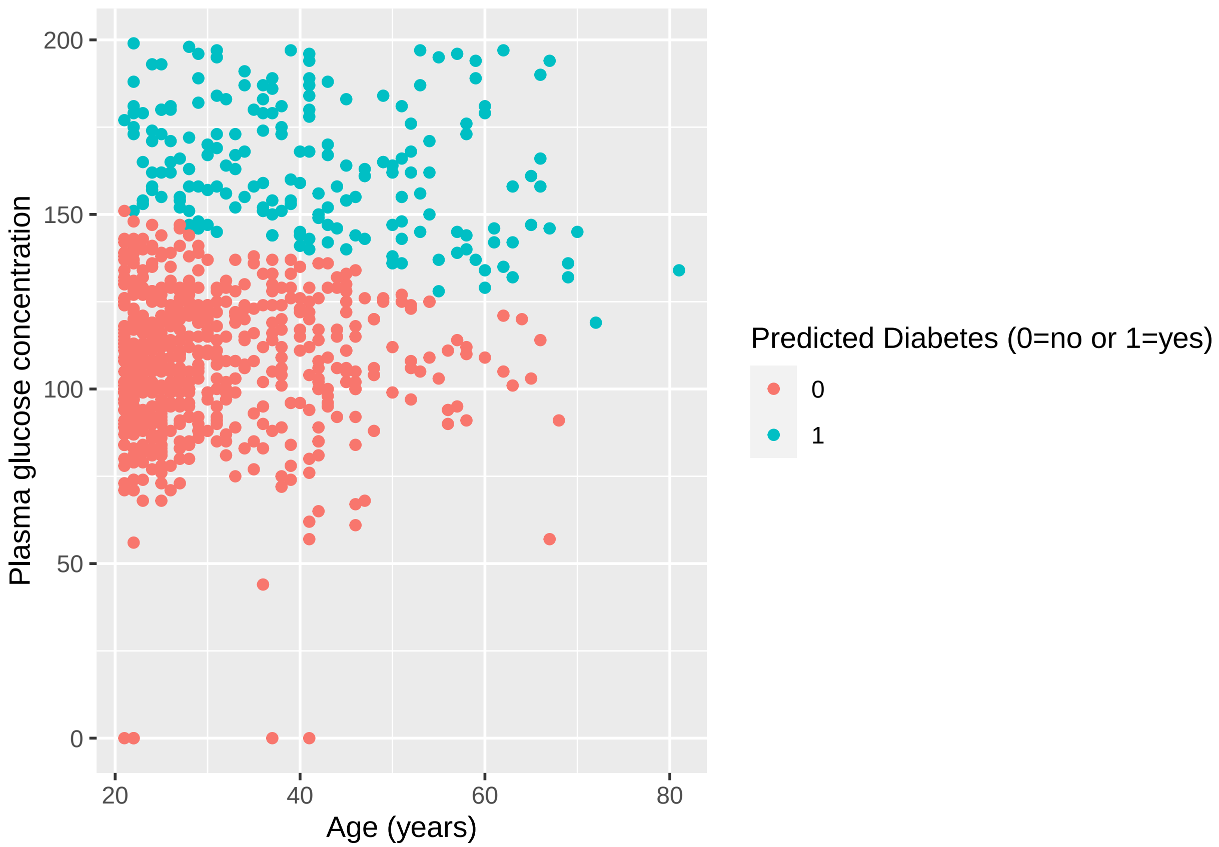
```
tq1 <- table(fit, data_PIdiabetes$X9)
tq1
```

```
##    
## fit   0   1
##   0 436 140
##   1  64 127
```

```
# confusion matrix
1 - (sum(diag(tq1)) / sum(tq1))
```

```
## [1] 0.2659713
```

```
# misclassification error



data_PIdiabetes_q1 <- data_PIdiabetes
data_PIdiabetes_q1$X10 <- factor(fit)

ggplot(data = data_PIdiabetes_q1, aes(x = X8, y = X2, colour = X10)) +
  geom_point() +
  labs(x = "Age (years)", y = "Plasma glucose concentration",
       colour = "Predicted Diabetes (0=no or 1=yes)" )
```



- From the glm model, Both Plasma glucose and Age have a significant influence on Diabetes. From the results of this classification, we can see that our misclassification error is about 26.6%. This is not too bad I think. Just as seen with the original plot earlier, the plot with the predicted Diabetes also depits that level-1 Diabetes have more plasma glucose compare to level-0 group

3.Use the model estimated in step 2 to a) report the equation of the decision boundary between the two classes b) add a curve showing this boundary to the scatter plot in step 2. Comment whether the decision boundary seems to catch the data distribution well.

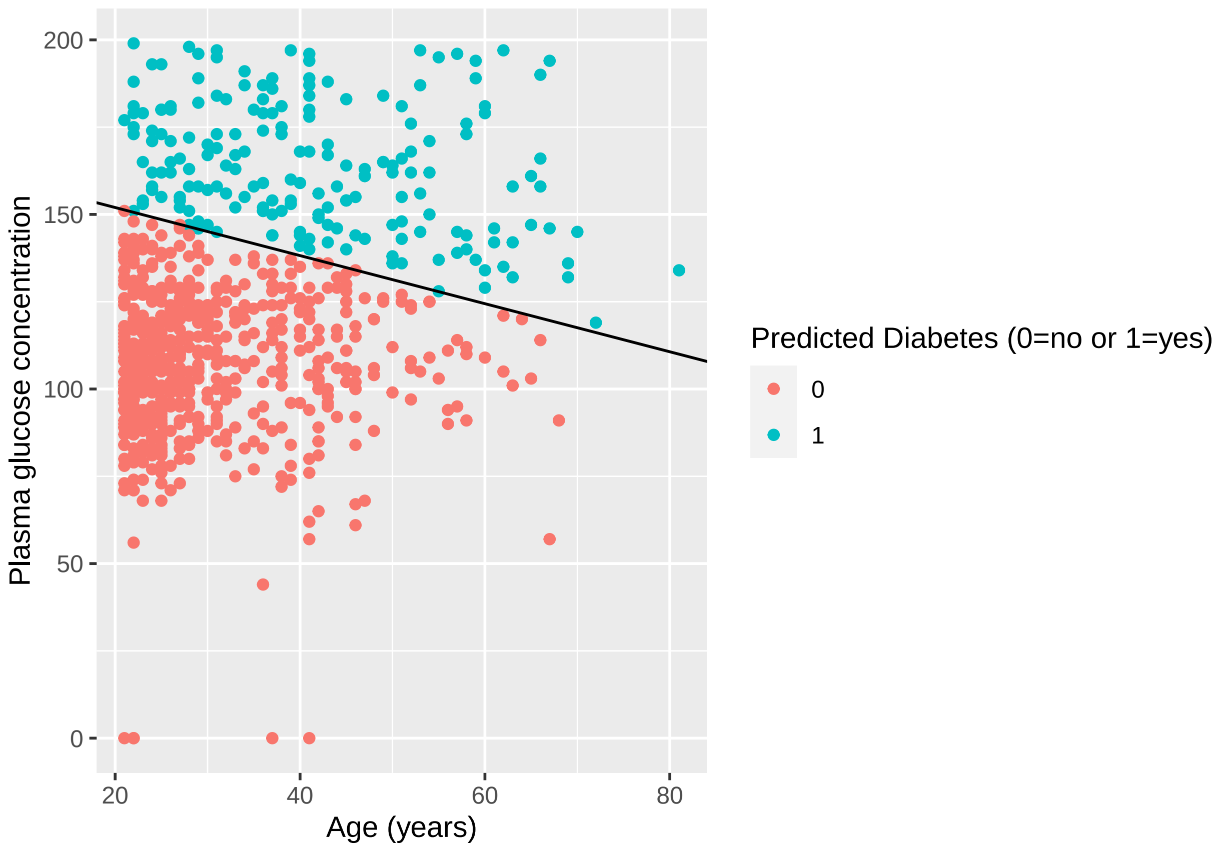
- when r = 0.5, we have

$$\frac{1}{1+e^{-\theta^T x}} = 0.5$$

then we have $\theta^T x = 0$, i.e. $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$ $(x_1 : concentration, x_2 : age)$

thus $slope = \frac{-\theta_2}{\theta_1}$, $intercept = \frac{-\theta_0}{\theta_1}$

```
theta <- coef(glm_diabete)

slope <- (-theta[3]) / theta[2]
intercept <- (-theta[1] ) / (theta[2])

ggplot(data = data_PIdiabetes_q1, aes(x = X8, y = X2, colour = X10)) +
  geom_point() +
  geom_abline(slope = slope, intercept = intercept) +

  labs(x = "Age (years)", y = "Plasma glucose concentration",
       colour  = "Predicted Diabetes (0=no or 1=yes)")
```



- Comments:I think this decision boundary does well for separating the data belonging two classes, even though not all data are put into the right sides of the boundary line.

4.Make same kind of plots as in step 2 but use thresholds r = 0.2 and r = 0.8. By using these plots, comment on what happens with the prediction when r value changes.

```
fit_r02 <- ifelse(fitted(glm_diabete) >= 0.2, 1, 0)

tr02 <- table(fit_r02, data_PIdiabetes$X9)
tr02
```

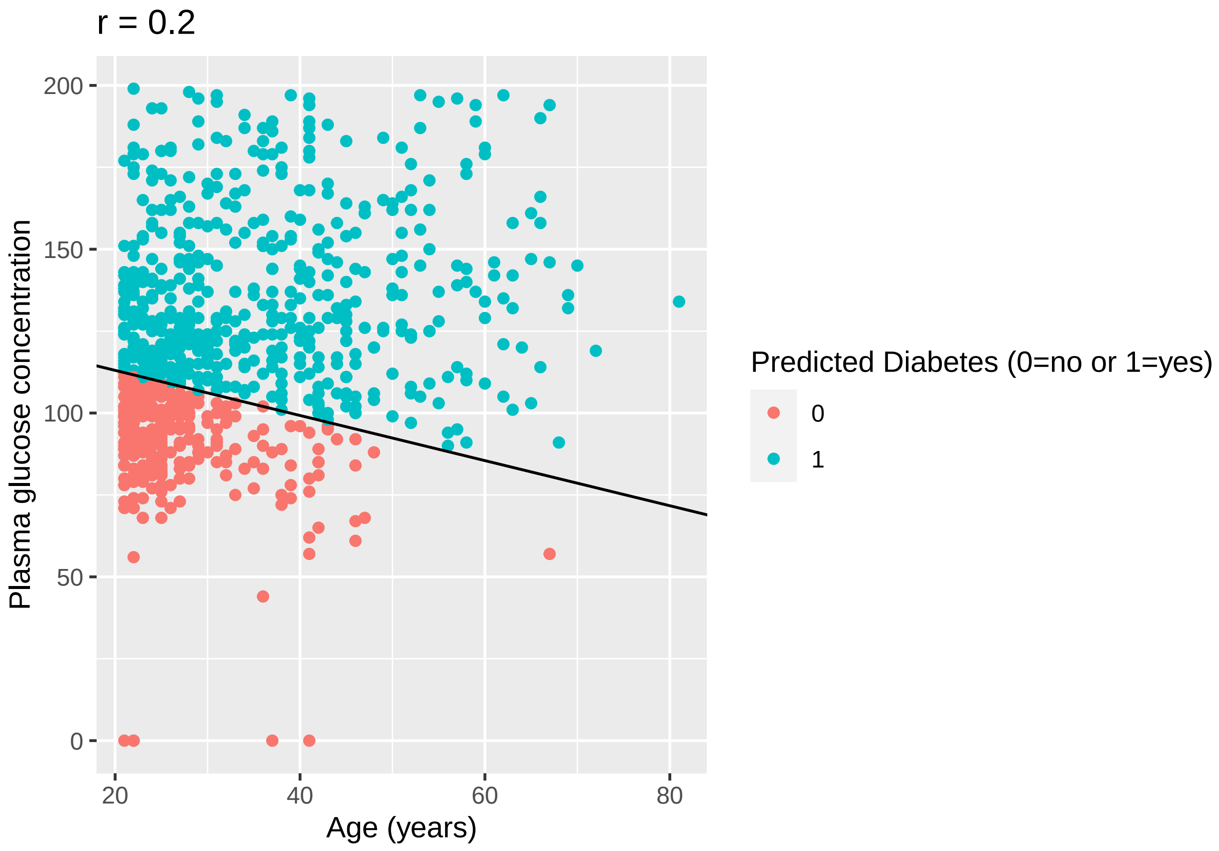
```
##
## fit_r02   0   1
##       0 238  25
##       1 262 242
```

```
#confusion matrix
1 - (sum(diag(tr02)) / sum(tr02))
```

```
## [1] 0.3741851
```

```
#misclassification error



data_PIdiabetes_r02 <- data_PIdiabetes
data_PIdiabetes_r02$X10 <- factor(fit_r02)

slope <- (-theta[3]) / theta[2]
intercept <- (-log(4) - theta[1] ) / (theta[2])
```

```
ggplot(data = data_PIdiabetes_r02, aes(x = X8, y = X2, colour = X10)) +
  geom_point() +
  geom_abline(slope = slope, intercept = intercept) +

  labs(x = "Age (years)", y = "Plasma glucose concentration",
       colour  = "Predicted Diabetes (0=no or 1=yes)", title = "r = 0.2" )
```



```
fit_r08 <- ifelse(fitted(glm_diabete) >= 0.8, 1, 0)


tr08 <- table(fit_r08, data_PIdiabetes$X9)
tr08
```

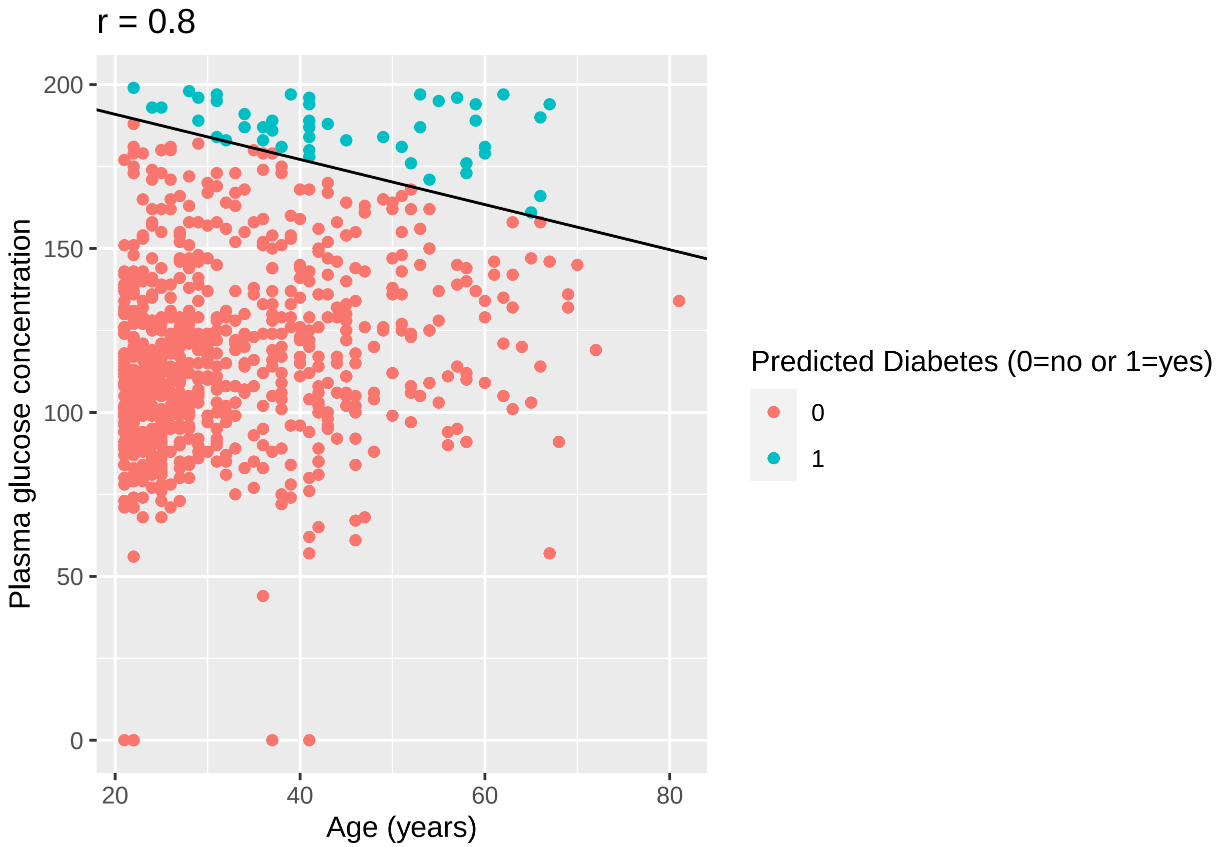
```
##
## fit_r08   0   1
##       0 490 231
##       1  10  36
```

```
#confusion matrix
1 - (sum(diag(tr08)) / sum(tr08))
```

```
## [1] 0.3142112
```

```
#misclassification error

data_PIdiabetes_r08 <- data_PIdiabetes
data_PIdiabetes_r08$X10 <- factor(fit_r08)
```

```r
slope <- (-theta[3]) / theta[2]
intercept <- (log(4) - theta[1] ) / (theta[2])

ggplot(data = data_PIdiabetes_r08, aes(x = X8, y = X2, colour = X10)) +
  geom_point() +
  geom_abline(slope = slope, intercept = intercept) +

  labs(x = "Age (years)", y = "Plasma glucose concentration",
       colour  = "Predicted Diabetes (0=no or 1=yes)", title = "r = 0.8" )
```



- Comment: When r is 0.8, we can find that the method is more likely to regard the patients with diabetes as healthy. The probability is 87%. By contrast, when we use r = 0.2 as the threshold, we can find the patients with more probability, which is 91%.

```r
242/(242+25)
```

```
## [1] 0.906367
```

```r
231/(231+36)
```

```
## [1] 0.8651685
```

5.Perform a basis function expansion trick by computing new features $z_1 = x_1^4, z_2 = x_1^3 x_2^1, z_3 = x_1^2 x_2^2, z_4 = x_1^1 x_2^3, z_5 = x_2^4$, adding them to the data set and then computing a logistic regression model with y as target and $x_1, x_2, z_1, \ldots, z_5$ as features. Create a scatterplot of the same kind as in step 2 for this model and compute the training misclassification rate. What can you say about the quality of this model compared to the previous logistic regression model? How have the basis expansion trick affected the shape of the decision boundary and the prediction accuracy?

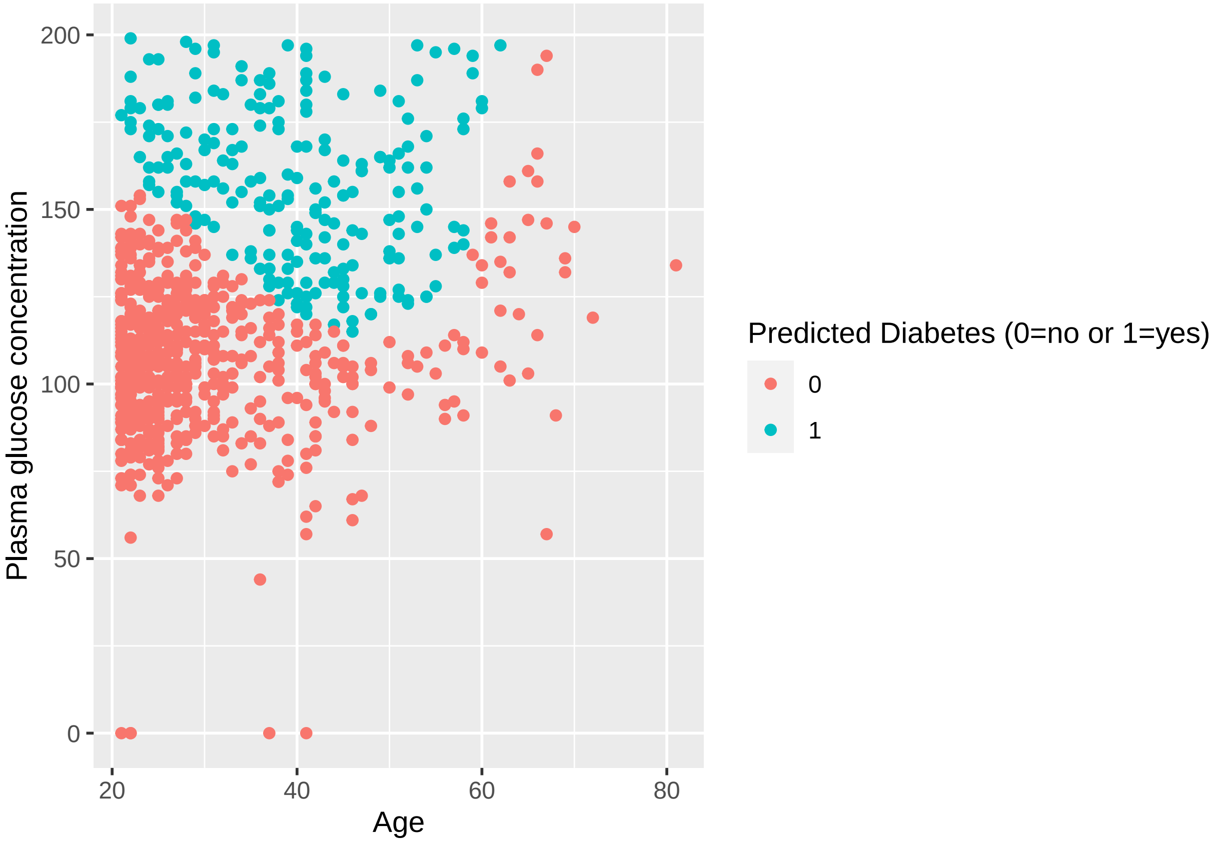
```
data_PIdiabetes_q5 <-  data_PIdiabetes %>%  select(c(X2,X8,X9))
# X2,X8,X9
# 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
# 8. Age (years).
# 9. Diabetes (0=no or 1=yes).

data_PIdiabetes_q5$Z1 <- data_PIdiabetes_q5$X2^4
data_PIdiabetes_q5$Z2 <- data_PIdiabetes_q5$X2^3 * data_PIdiabetes_q5$X8^1
data_PIdiabetes_q5$Z3 <- data_PIdiabetes_q5$X2^2 * data_PIdiabetes_q5$X8^2
data_PIdiabetes_q5$Z4 <- data_PIdiabetes_q5$X2^1 * data_PIdiabetes_q5$X8^3
data_PIdiabetes_q5$Z5 <- data_PIdiabetes_q5$X8^3

glm_diabete_q5 <- glm(X9 ~ X2 + X8 + Z1 + Z2 + Z3 + Z4 + Z5, data_PIdiabetes_q5, family = binomial)

fit <- ifelse(fitted(glm_diabete_q5) >= 0.5, 1, 0)

tq5 <- table(fit, data_PIdiabetes_q5$X9)
tq5
```

```
##
## fit   0   1
##   0 434 118
##   1  66 149
```

```
# confusion matrix
1 - (sum(diag(tq5)) / sum(tq5))
```

```
## [1] 0.2398957
```

```
# misclassification error

data_PIdiabetes_q5$X10 <- factor(fit)

ggplot(data = data_PIdiabetes_q5, aes(x = X8, y = X2, colour = X10)) +
  geom_point() +
  labs(x = "Age", y = "Plasma glucose concentration",
       colour  = "Predicted Diabetes (0=no or 1=yes)")
```



- Comment: When we increase the number of inputs and r equals 0.5, the eventual value of misclassification rate is similar to the one in the second question. The boundary line is no longer a straight line but a curve.