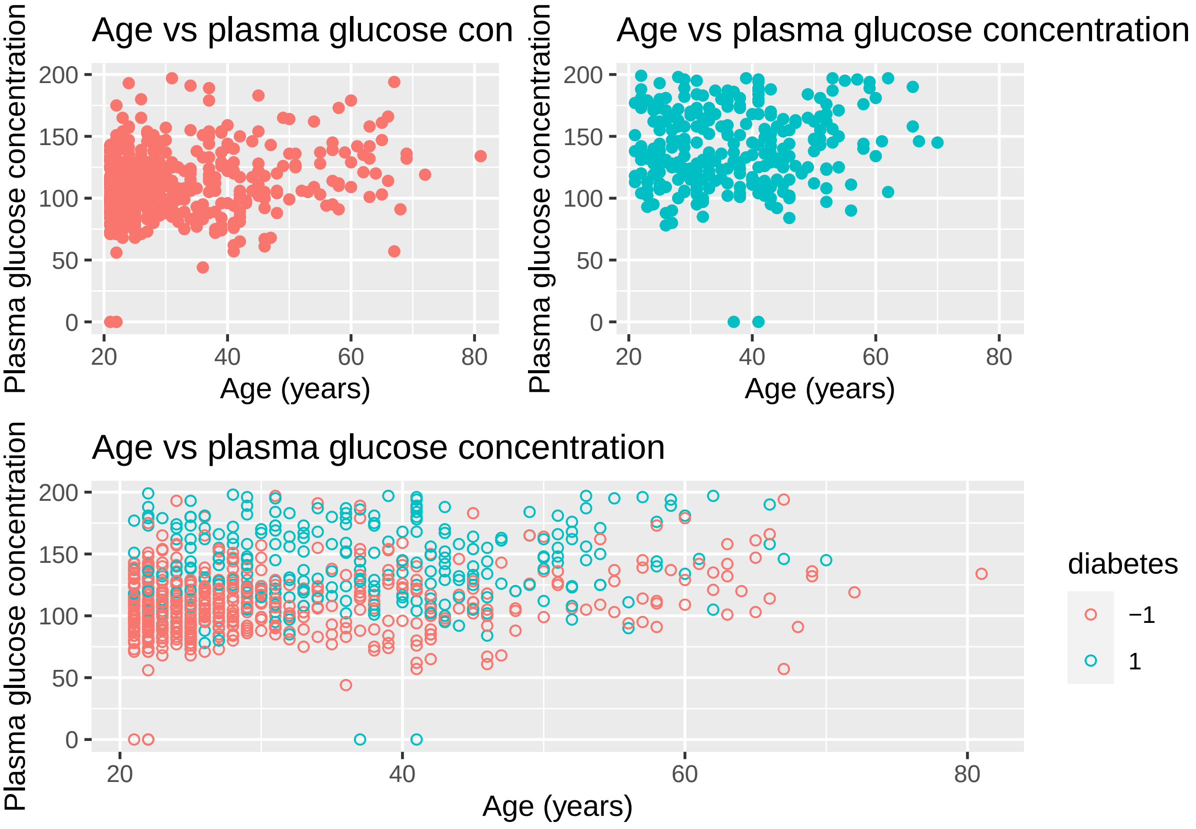
Lab01 report

2022-11-08

Assignment 3: Logistic regression

3.1 Plot

Plot of age vs glucose plasma concentration, colour by diabetes:



There appears to be considerable overlap between observations classified as diabetes and observations classified as non-diabetes, but it is not so easy from the plot to determine the extent of the overlap. However, it seems it will not be very easy to classify the observations into the two groups.
It can be noted that there are some observations with very low/zero plasma glucose levels, these observations may be erroneous.

3.2: Logistic regression model

A logistic regression model with diabetes as target and glucose plasma concentration and age as features were performed.

Estimated regression coefficients:

```
#> (Intercept)  glucose_pl         age
#> -5.91244906  0.03564404  0.02477835
```

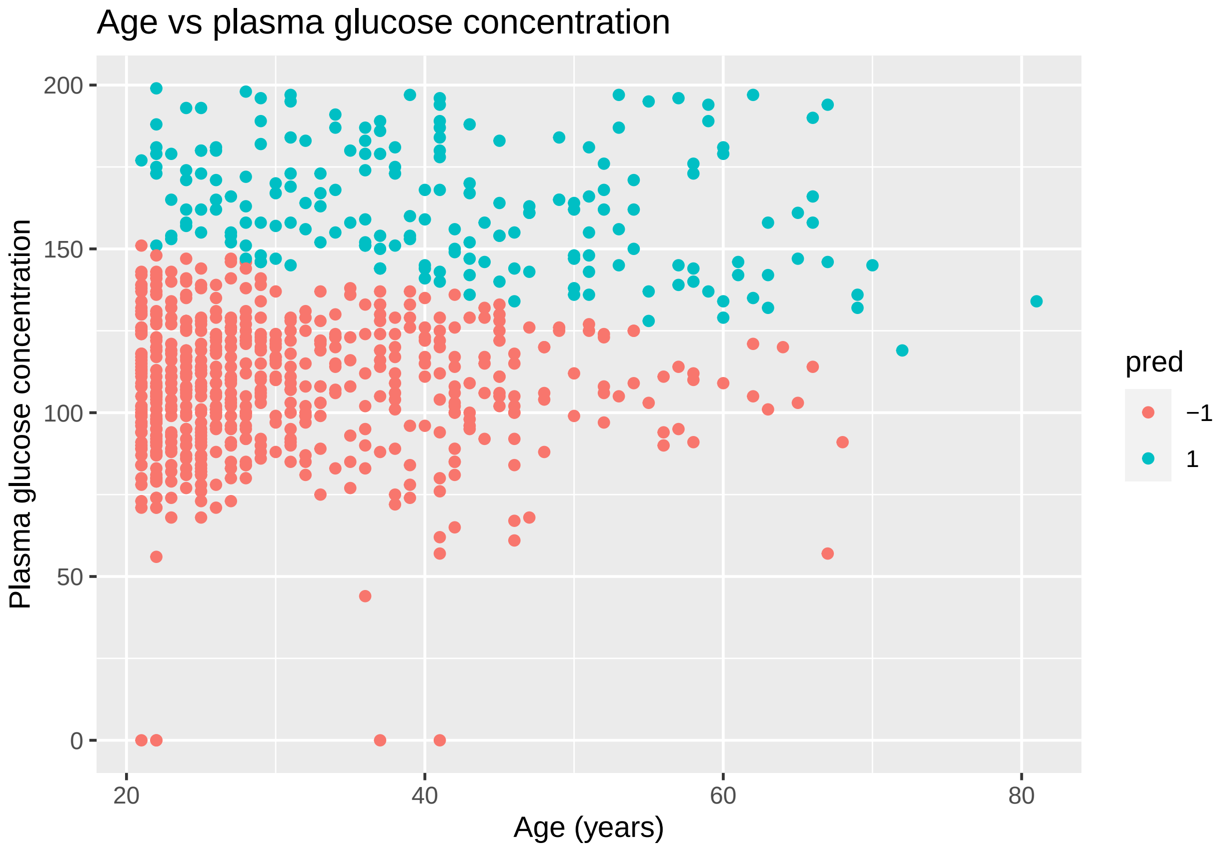
Probabilistic model:

$$p(y=1|\boldsymbol{x}) = g(\boldsymbol{x}) = \frac{e^{\boldsymbol{\theta}^T\boldsymbol{x}}}{1+e^{\boldsymbol{\theta}^T\boldsymbol{x}}} = \frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} = \frac{1}{1+e^{-(-5.91244906+0.03564404x_1+0.02477835x_2)}}$$

Training misclassification error (percent) and confusion matrix:

```
#> Confusion matrix:

#>       pred
#> target  -1   1
#>     -1 436  64
#>     1  138 130

#> Misclassification error:

#> [1] 26.3

#> Percentage of observations that is non-diabetes:

#> [1] 65.1
```

Plot of age vs glucose plasma concentration, colour by predicted values:



The prediction error is quite high, 26%, while it means that 74% of observations are correctly classified. However, in assessing the discriminative ability of the model one need to consider the imbalance in the number of observations in the different classes, with 65% of the observations belonging to the non-diabetes class. The misclassification rate is larger for the smaller group.

3.3 Decision boundary

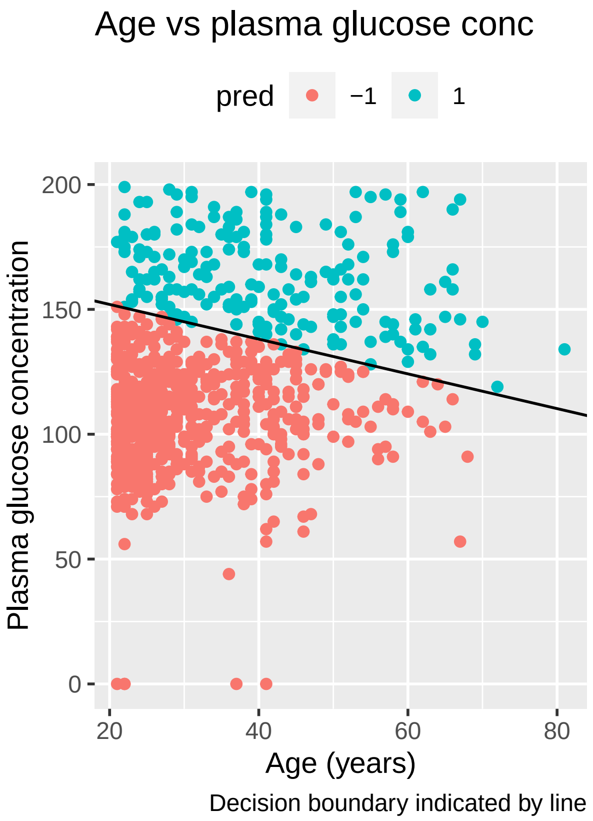
Decision boundary equation:
Setting

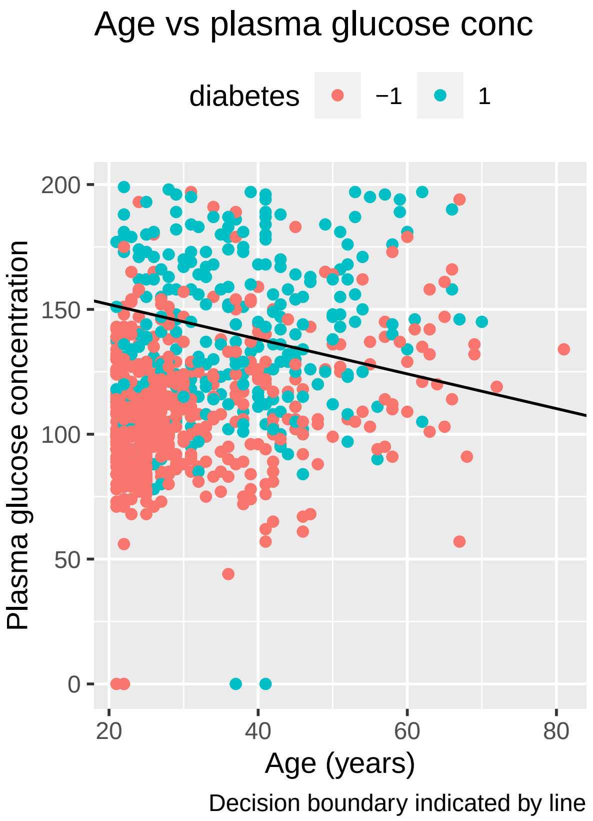
$$g(\boldsymbol{x}) = 1 - g(\boldsymbol{x})$$

gives

$$x_1 = \frac{\theta_0}{\theta_1} - \frac{\theta_2}{\theta_1} * x_2$$

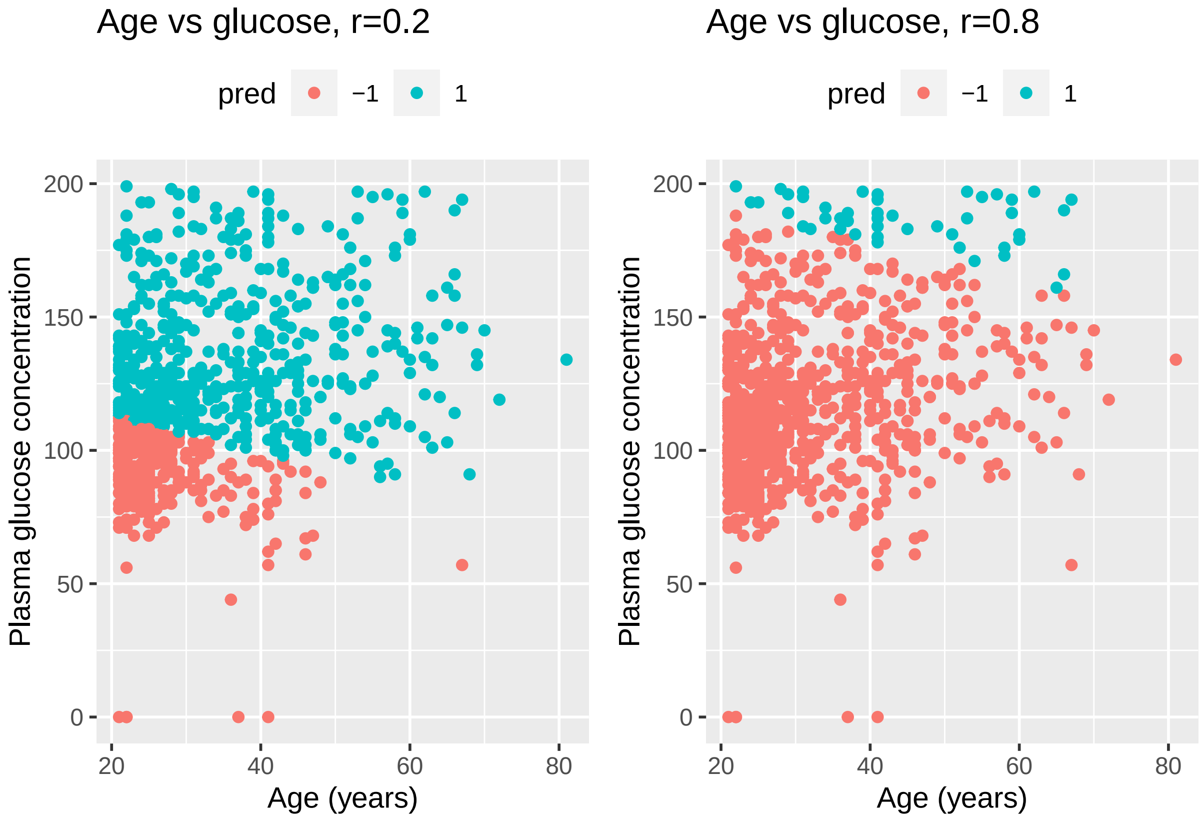
Plot of data with decision boundary:





The decision boundary appears to be in the center of the smaller group (diabetes), so not really between the groups.

3.4 Plots with thresholds 0.2 and 0.8



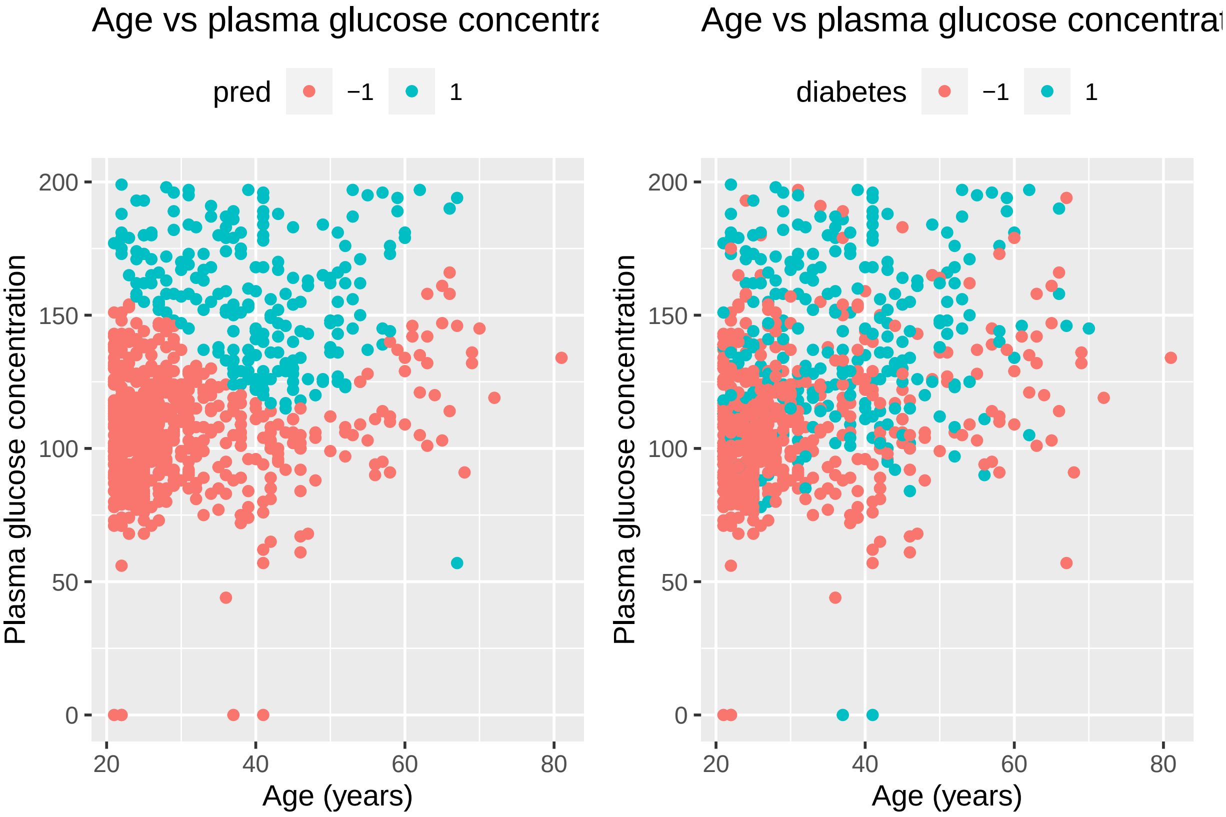
```
#> Confusion matrix r=0.2:

#>        pred
#> target  -1   1
#>     -1 238 262
#>     1   24 244

#> Confusion matrix r=0.8:

#>        pred
#> target  -1   1
#>     -1 490  10
#>     1  232  36

#> Missclassification rate r=0.2:

#> [1] 37.24

#> Missclassification rate r=0.8:

#> [1] 31.51
```

With r = 0.2 compared to using r = 0.5, more observations are being classified as diabetes and the misclassification error for the diabetes group is lower, however the overall misclassification error is higher.

With r = 0.8 compared to using r = 0.5, more observations are being classified as non-diabetes, however the overall misclassification error is higher. Compared to using r = 0.2, the misclassification error is lower.

3.5 Basis function expansion

Scatter plot:



Misclassification rate and confusion matrix:

```
#> Confusion matrix:
```

```
#>       pred
#> target  -1   1
#>     -1 433  67
#>     1  121 147
```

```
#> Misclassification rate (%):
```

```
#> [1] 24.48
```

The misclassification error for the training set is now 24%, which is an improvement compared to the previous model. However, it remains to be seen how the respective models would perform on a validation data set to determine which one is better. With the new model, there is an indication as seen in the plot that individuals of higher age but with low plasma glucose concentration would be classified as belonging to the diabetes group, which might not be a successful classification. This latter property may be possible to remove if one excludes the observations with plasma glucose $= 0$, if these measurements can be shown to be erroneous. Code that investigates the influence of these observations is in the the Appendix. However, it may be that even after removal of any erroneous values, the new model could prove to be too sensitive to data points that are deviating due to noise (and not that they are erroneous).

The decision boundar(ies) are no longer linear, and further there is more than one boundary.

Appendix: All code for this report

```
knitr::opts_chunk$set(comment = "#>",
                       echo = FALSE)
#------------------------------------------------------------------------------#
#- Set libraries
#------------------------------------------------------------------------------#
library(tidyverse)
library(caret)
library(ggplot2)
library(patchwork)
library(scales)
library(tinytex)
#------------------------------------------------------------------------------#
#- 0) Read in data and divide into training, validation and test
#------------------------------------------------------------------------------#

file_in <- "C:/Users/kerstin/Documents/LiU/ML/Labs/Lab01/Data/pima-indians-diabetes.csv"
data_in <- read_csv(file_in, col_names = FALSE)

spec(data_in)

names(data_in) <- c(
  "pregnant_num",
  "glucose_pl",
  "bp_dia",
  "triceps_skin_thick",
  "insulin_serum",
  "bmi",
  "diabetes_pedigree",
  "age",
  "diabetes")

#- Set factors
data_1 <- mutate(data_in,diabetes=ifelse(diabetes<1,-1,1),
                 diabetes=as.factor(diabetes))

data <- data_1

#------------------------------------------------------------------------------#
#- 1) Make a plot of age vs plasma glucose concentration
#------------------------------------------------------------------------------#
ylim_min <- min(data$glucose_pl)
ylim_max <- max(data$glucose_pl)

xlim_min <- min(data$age)
xlim_max <- max(data$age)

#- Obtain colour codes for plot
hex <- hue_pal()(2)

p1 <- data %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+ geom_point(shape=21, fill = NA) +
  labs(title="Age vs plasma glucose concentration",
```

```r
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + xlim(xlim_min, xlim_max)

p2 <- data %>% filter(diabetes==-1) %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+
  geom_point(colour= "#F8766D") +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + xlim(xlim_min, xlim_max)

p3 <- data %>% filter(diabetes==1) %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+
  geom_point(colour= "#00BFC4") +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + xlim(xlim_min, xlim_max)

(p2 + p3)/p1

#- There are observations with glucose levels = 0, identify those
order_glucose <- order(data$glucose_pl)
data$glucose_pl[order_glucose[1:10]]
obs_with_glucose_eq_0 <- which(data$glucose_pl==0)

#-----------------------------------------------------------------------------#
#- 2) Train a logistic regression model with y = Diabetes as target and
# x1 = Plasma glucose concentration and x2 = Age as features and make a
# prediction for all observations by using r = 0.5 as the classification
# threshold
#-----------------------------------------------------------------------------#

train <- data%>%select(glucose_pl,age,diabetes)

m1 <- glm(diabetes~., train, family = "binomial")

#- Regression coefficients:
(coef(m1))


#- Model with observations with plasma glucose excluded
train_b <- dplyr::filter(train, glucose_pl > 0)

m1_b <- glm(diabetes~., train_b, family = "binomial")

#- Regression coefficients:
(coef(m1_b))

#- Training misclassification error (percent) and confusion matrix
prob <- predict(m1, type="response")
pred <- ifelse(prob>0.5, 1, -1)

cat("Confusion matrix:")
(cm_train <- table(train$diabetes, pred, dnn=c("target","pred")))

cat("Misclassification error:")
```

```
(err_train_perc <- round(100*((dim(train)[1]-sum(diag(cm_train)))/dim(train)[1]),2))

#cat("Number of observations in the different groups:")
no_obs_grp <- table(train$diabetes)

no_obs_grp_b <- table(train_b$diabetes)

cat("Percentage of observations that is non-diabetes:")
(round(100*unname(no_obs_grp[1]) /sum(no_obs_grp),2))

#- Model excluding obs with glucose = 0
prob_b <- predict(m1_b, type="response")
pred_b <- ifelse(prob_b>0.5, 1, -1)

(cm_train_b <- table(train_b$diabetes, pred_b, dnn=c("target","pred")))

(err_train_perc_b <- round(100*((dim(train_b)[1]-sum(diag(cm_train_b)))/dim(train_b)[1]),2))


#- Plot of age vs glucose plasma concentration, colour by predicted values:
data_plot <- add_column(data,pred)

data_plot <- mutate(data_plot,pred=as.factor(pred))

ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)

p1 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max)

p1

#-----------------------------------------------------------------------------#
# 3) Use the model estimated in 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
#-----------------------------------------------------------------------------#

#- Calculate intercept and slope for the decision boundary
theta_0 <- coef(m1)[1]
theta_v <- as.matrix(coef(m1)[-1])

intercept <- -theta_0/theta_v[1]
slope <- -theta_v[2]/theta_v[1]

ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)

#- Plot decision boundary  ----#

#- Colour by pred
```

```
p1 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  geom_abline(slope=slope, intercept = intercept, show.legend = TRUE) +
  labs(title="Age vs plasma glucose conc",
       y="Plasma glucose concentration", x="Age (years)", caption =
         "Decision boundary indicated by line") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

#- Colour by target
p2 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+ geom_point() +
  geom_abline(slope=slope, intercept = intercept, show.legend = TRUE) +
  labs(title="Age vs plasma glucose conc",
       y="Plasma glucose concentration", x="Age (years)", caption =
         "Decision boundary indicated by line") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

(p1+p2)
#-----------------------------------------------------------------------------#
# 4) Make same kind of plots as in 2) but use thresholds r = 0.2 and r = 0.8.
#-----------------------------------------------------------------------------#
#- r: 0.2
pred <- ifelse(prob>0.2, 1, -1)

#- Calculate confusion matrix and misclassification error
cm_train_0_2 <- table(train$diabetes, pred, dnn=c("target","pred"))
err_train_perc_0_2 <- round(100*((dim(train)[1]-sum(diag(cm_train_0_2)))/dim(train)[1]),2)

#- Make plot
data_plot <- add_column(data,pred)

data_plot <- mutate(data_plot,pred=as.factor(pred))

ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)

p1_0_02 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  labs(title="Age vs glucose, r=0.2",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

#- r: 0.8
pred=ifelse(prob>0.8, 1, -1)

#- Calculate confusion matrix and misclassification error
cm_train_0_8 <- table(train$diabetes, pred, dnn=c("target","pred"))
err_train_perc_0_8 <- round(100*((dim(train)[1]-sum(diag(cm_train_0_8)))/dim(train)[1]),2)

#- Make plot
data_plot <- add_column(data,pred)

data_plot <- mutate(data_plot,pred=as.factor(pred))

ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)
```

```
p1_0_08 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  labs(title="Age vs glucose, r=0.8",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

(p1_0_02 + p1_0_08)

#- Print confusion matrices and misclassification errors
cat("Confusion matrix r=0.2:")
(cm_train_0_2)
cat("Confusion matrix r=0.8:")
(cm_train_0_8)

cat("Missclassification rate r=0.2:")
(err_train_perc_0_2)
cat("Missclassification rate r=0.8:")
(err_train_perc_0_8)
#------------------------------------------------------------------------------#
# 5) Perform a basis function expansion trick by computing new features
# z1 = x1^4, z2 = x1^3*x2, z3= x1^2*x2^2, z4 = x1*x2^3, z5=x2^4, adding them
# to the data set and then computing a logistic regression model with y as target
# and x1,x2,z1, ... z5 as features. Create a scatterplot of the same kind as in
# 2) for this model and compute the training misclassification rate.
#------------------------------------------------------------------------------#

#- Compute new variables and train model
data_2 <- data %>% mutate(z1=glucose_pl^4,
                          z2=glucose_pl^3*age,
                          z3=glucose_pl^2*age^2,
                          z4=glucose_pl*age^3,
                          z5=age^4)


train <- data_2 %>%select(glucose_pl,age,diabetes,z1:z5)

m2 <- glm(diabetes~., train, family = "binomial")
prob <- predict(m2, type="response")
pred <- ifelse(prob>0.5, 1, -1)

#- Model without observations with glucose = 0
data_2_b <- filter(data_2, glucose_pl> 0)
train_b <- data_2_b %>%select(glucose_pl,age,diabetes,z1:z5)

m2_b <- glm(diabetes~., train_b, family = "binomial")
prob_b <- predict(m2_b, type="response")
pred_b <- ifelse(prob_b>0.5, 1, -1)

#- Plot of data

data_plot <- add_column(data,pred)

data_plot <- mutate(data_plot,pred=as.factor(pred))
```

```
ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)

p1 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

p2 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+ geom_point() +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max)+ theme(legend.position="top")

(p1 + p2)


#- Print confusion matrix and misclassification rate
cat("Confusion matrix:")
(cm_train_2 <- table(train$diabetes, pred, dnn=c("target","pred")))

cat("Misclassification rate (%):")
(err_train_perc_2 <- round(100*((dim(train)[1]-sum(diag(cm_train_2)))/dim(train)[1]),2))
#- Model without observations with plasma glucose = 0 ------------#
#- Plot of data

data_b <- filter(data, glucose_pl > 0)
data_plot <- add_column(data_b,pred_b)

data_plot <- mutate(data_plot,pred=as.factor(pred_b))

ylim_min <- min(data_plot$glucose_pl)
ylim_max <- max(data_plot$glucose_pl)

p1 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=pred)+ geom_point() +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max) + theme(legend.position="top")

p2 <- data_plot %>% ggplot() + aes(x=age,y=glucose_pl, colour=diabetes)+ geom_point() +
  labs(title="Age vs plasma glucose concentration",
       y="Plasma glucose concentration", x="Age (years)") +
  ylim(ylim_min, ylim_max)+ theme(legend.position="top")

(p1 + p2)

#- Print confusion matrix and misclassification rate

cat("Confusion matrix:")
(cm_train_2_b <- table(train_b$diabetes, pred_b, dnn=c("target","pred")))

cat("Misclassification rate (%):")
(err_train_perc_2_b <- round(100*((dim(train_b)[1]-sum(diag(cm_train_2_b)))/dim(train_b)[1]),2))
```