Lab 3 2020-02-07 V1.01 - Exercise answers

Biomedical Data Science

Question 1

Fit a logistic regression for CHD and age and compute odds ratios and confidence interval:

```
> chdage <- read.csv("data/chdagesex.csv")
> fit.m1 <- glm(CHD ~ AGE, data=chdage, family="binomial")
> or.age <- exp(coef(fit.m1)[2])
> ci.age <- exp(confint(fit.m1))[2, ]
> round(c(or.age, ci.age), 2)
   AGE  2.5 % 97.5 %
  1.06   1.04   1.09
```

Aside - classroom question

What is the odds ratio or probability at 0 years or any other age ?

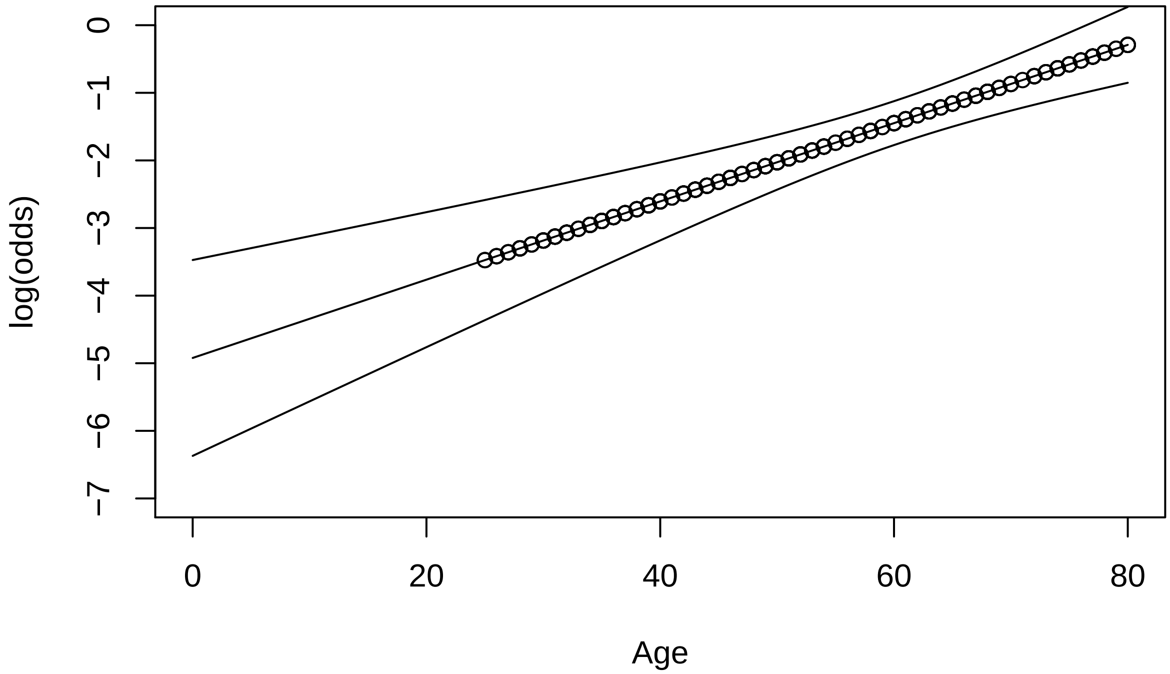
Since the range of AGE in the dataset does not include any age values at or close to 0 years, then the confidence interval will likely be large when calculating the odds at lower ages.

```
> summary(chdage$AGE)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  25.00   39.00   51.00   51.93   65.00   80.00
> summary(fit.m1)

Call:
glm(formula = CHD ~ AGE, family = "binomial", data = chdage)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0568  -0.6206  -0.4218  -0.2993   2.5834

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.92170    0.73898  -6.660 2.74e-11 ***
AGE          0.05789    0.01192   4.855 1.20e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 263.80  on 299  degrees of freedom
Residual deviance: 236.08  on 298  degrees of freedom
AIC: 240.08

Number of Fisher Scoring iterations: 5
```

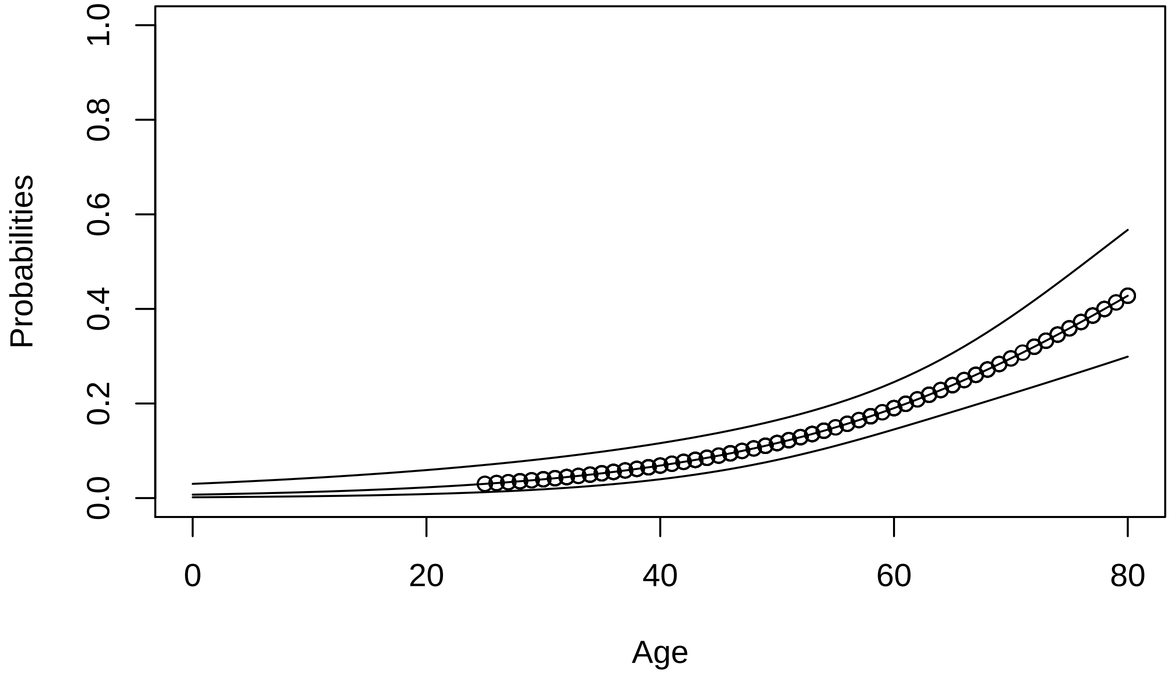
From the `summary(fit.m1)` we see that the log-odds calculated at 1 year is 0.0578887

```
> require(data.table)
>
> fit.m1.logodds <- summary(fit.m1)$coefficients[2]
```

```
>
> # Plot the absolute predicted probabilities for ages 0 to 80.
> age.range <- 0:80
> fit.predicted <- predict(fit.m1, data.frame(AGE=age.range), se.fit=T)
> fit.predicted.dt <- data.table(AGE=age.range, fit=fit.predicted$fit)
>
> # Prove the predicted probability at age 0 is as per the intercept:
> at.zero <- as.numeric(fit.predicted.dt[AGE==0]$fit)
> # Is the intercept (converting to odds and then probability)
> at.intercept <- as.numeric(fit.m1$coefficients[1])
> all.equal(at.intercept, at.zero)
[1] TRUE
>
> # So, if we want to manually calculate the probability at age 40, we need to take
> # the 1 year log odds, multiply by 40 and add the intercept. We then convert the
> # result to odds and finally to probabilities:
>
> # Calculate the probability at 40 years of age
> at.40 <- exp(fit.m1.logodds * 40 + as.numeric(fit.m1$coefficients[1]))
> # y = mx + c
> at.40.prob <- at.40/(1 + at.40)
> at.40.prob
[1] 0.0687438
> # Combine the original dataset with the model fitted values
> chdage.fitted <- data.table(cbind(chdage,fit.m1$fitted.values))
> # Show the fitted values from the model at age 40
> chdage.fitted[chdage.fitted$AGE==40]
   AGE SEX CHD fit.m1$fitted.values
1:  40   M   0            0.0687438
2:  40   M   1            0.0687438
3:  40   F   0            0.0687438
4:  40   F   0            0.0687438
5:  40   M   0            0.0687438
6:  40   F   0            0.0687438
>
>
> # Plot the log-odds from the model
> plot(chdage$AGE, fit.m1$linear.predictors,
+      xlim=c(min(age.range),max(age.range)), ylim=c(-7, 0),
+      xlab="Age", ylab="log(odds)")
>
> # Now plot the predicted range from 0 to 70
> lines(age.range, fit.predicted$fit)
> # And confidence intervals
> ci.upper.logodds <- fit.predicted$fit + fit.predicted$se.fit*1.96
> ci.lower.logodds <- fit.predicted$fit - fit.predicted$se.fit*1.96
> lines(age.range, ci.upper.logodds)
> lines(age.range, ci.lower.logodds)
```



```
>
>
> # Convert to probabilities - this scale doesn't give a clear indication
> # of the prediction confidence lower in the age range.
> plot(chdage$AGE, fit.m1$fitted.values,
+      xlim=c(min(age.range),max(age.range)), ylim=c(0, 1),
+      xlab="Age", ylab="Probabilities")
>
> # Convert the predicted values to probabilities and plot
> lines(age.range, exp(fit.predicted$fit) / (1 + exp(fit.predicted$fit)))
>
> # Add 95% confidence intervals
> ci.upper.odds <- exp(ci.upper.logodds)
> ci.upper.prob <- ci.upper.odds/(1+ci.upper.odds)
>
> ci.lower.odds <- exp(ci.lower.logodds)
> ci.lower.prob <- ci.lower.odds/(1+ci.lower.odds)
>
> lines(age.range, ci.upper.prob)
> lines(age.range, ci.lower.prob)
```



End of classroom question aside

Fit a logistic regression for CHD adjusted for age and sex:

```
> fit.m2 <- glm(CHD ~ AGE + SEX, data=chdage, family="binomial")
> or.age <- exp(coef(fit.m2)[2])
> ci.age <- exp(confint(fit.m2))[2, ]
> round(c(or.age, ci.age), 2)
   AGE  2.5 % 97.5 %
  1.06   1.04   1.09
> or.sex <- exp(coef(fit.m2)[3])
> ci.sex <- exp(confint(fit.m2))[3, ]
> round(c(or.sex, ci.sex), 2)
  SEXM  2.5 % 97.5 %
  2.62   1.34   5.32
```

Likelihood ratio test:

```
> pval <- pchisq(fit.m1$deviance - fit.m2$deviance, df=1, lower.tail=FALSE)
> signif(pval, 2)
[1] 0.0047
```

Given that the p-value is < 0.05, the model that includes sex is significantly better.

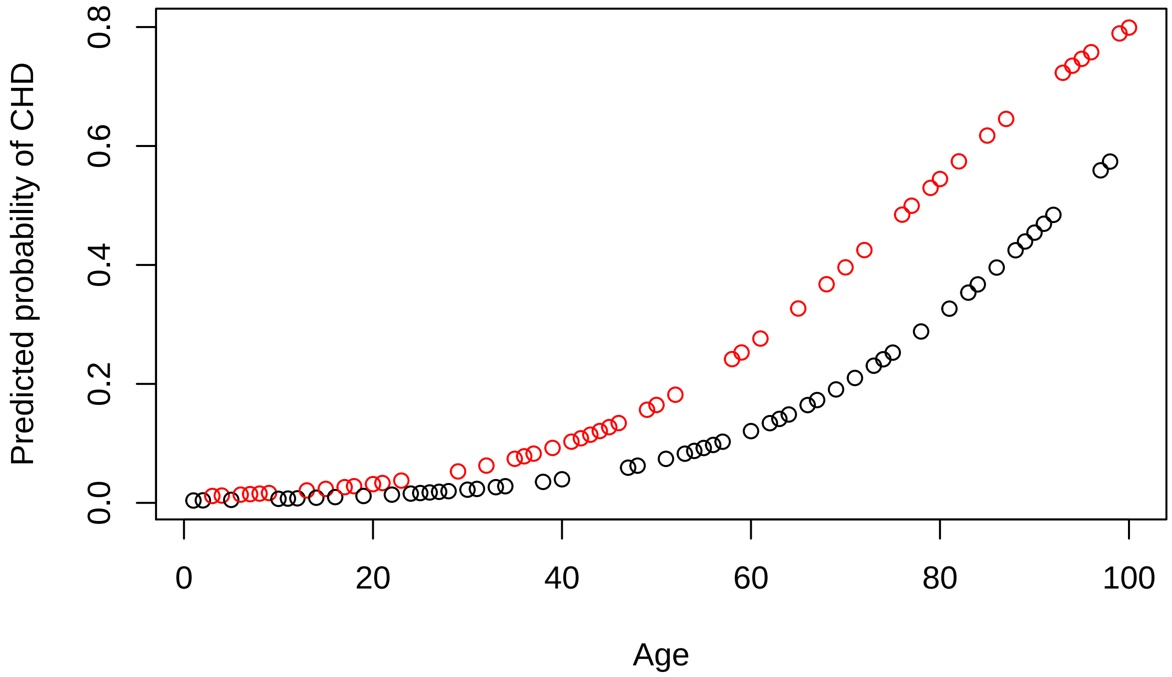
Stratification by age:

```
> regr.le50 <- glm(CHD ~ SEX, data=chdage, subset=AGE <= 50, family="binomial")
> or.le50 <- exp(coef(regr.le50)[2])
> ci.le50 <- exp(confint(regr.le50))[2, ]
> round(c(or.le50, ci.le50), 2)
  SEXM  2.5 % 97.5 %
  0.62   0.12   2.62
> regr.gt50 <- glm(CHD ~ SEX, data=chdage, subset=AGE > 50, family="binomial")
> or.gt50 <- exp(coef(regr.gt50)[2])
> ci.gt50 <- exp(confint(regr.gt50))[2, ]
> round(c(or.gt50, ci.gt50), 2)
  SEXM  2.5 % 97.5 %
  3.90   1.81   8.88
```

The stratified model shows that the relationship between sex and CHD exists only among older people: being a male over 50 has much higher odds of having a CVD event with respect to being a female in the same age group, but the same cannot be said for the younger group.

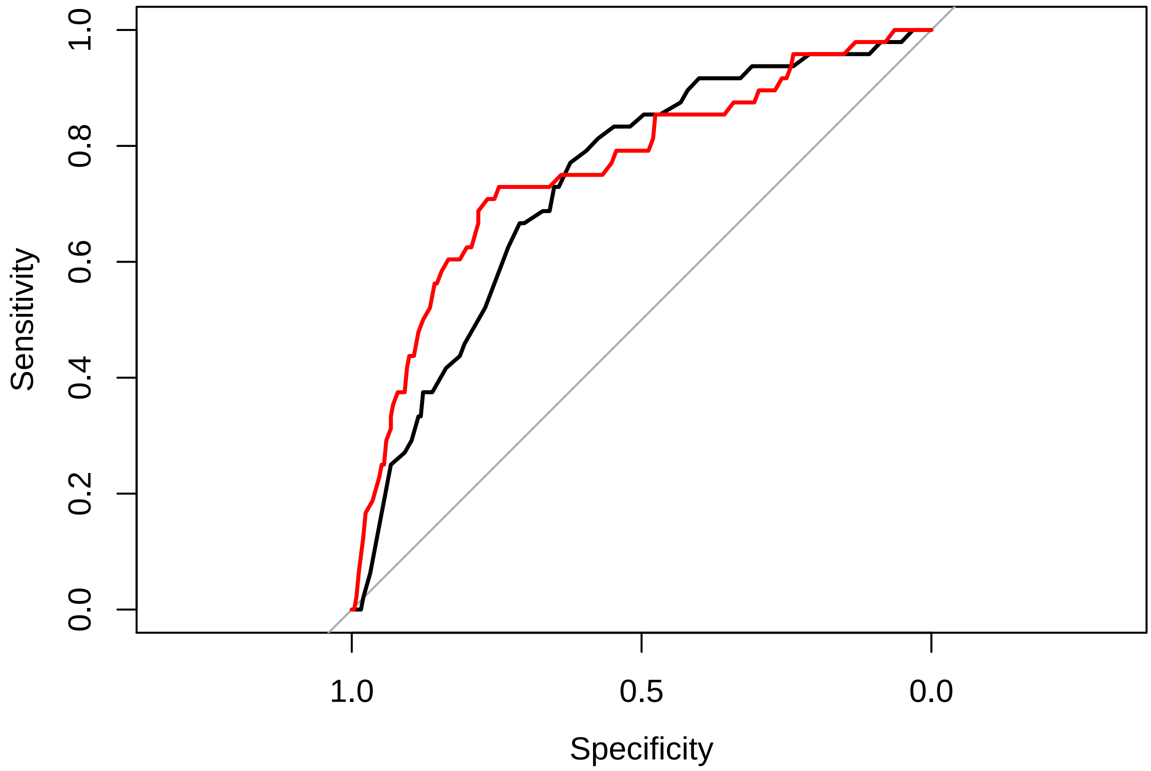
Create dataframe agesex and plot predicted probabilities:

```
> set.seed(1)
> agesex <- data.frame(AGE=1:100,
+                      SEX=factor(rbinom(100, 1, 0.5), labels=c("F", "M")))
> pred <- predict(fit.m2, type="response", newdata=agesex)
> plot(agesex$AGE, pred, xlab="Age", ylab="Predicted probability of CHD",
+      col=agesex$SEX)
```



ROC curves of the two models:

```
> library(pROC)
> roc(chdage$CHD, fit.m1$fitted.values, plot=TRUE)

Call:
roc.default(response = chdage$CHD, predictor = fit.m1$fitted.values,     plot = TRUE)

Data: fit.m1$fitted.values in 252 controls (chdage$CHD 0) < 48 cases (chdage$CHD 1).
Area under the curve: 0.734
> roc(chdage$CHD, fit.m2$fitted.values, plot=TRUE, add=TRUE, col="red")
```



```
Call:
roc.default(response = chdage$CHD, predictor = fit.m2$fitted.values,     plot = TRUE, add = TRUE, col =

Data: fit.m2$fitted.values in 252 controls (chdage$CHD 0) < 48 cases (chdage$CHD 1).
Area under the curve: 0.7601
```

Function `glm.cv()`:

```
> glm.cv <- function(formula, data, folds) {
+   regr.cv <- NULL
+   for (fold in 1:length(folds)) {
+     regr.cv[[fold]] <- glm(formula, data=data[-folds[[fold]], ],
+                            family="binomial")
+   }
+   return(regr.cv)
+ }
```

Run 10-fold cross-validation:

```
> library(caret)
> set.seed(1)
> folds <- createFolds(chdage$CHD, k=10)
> cv.m1 <- glm.cv(CHD ~ AGE, chdage, folds)
> cv.m2 <- glm.cv(CHD ~ AGE + SEX, chdage, folds)
```

Function `predict.cv()`:

```
> predict.cv <- function(regr.cv, data, outcome, folds) {
+   pred.cv <- NULL
+   for (fold in 1:length(folds)) {
+     test.idx <- folds[[fold]]
+     pred.cv[[fold]] <- data.frame(obs=outcome[test.idx],
+                                   pred=predict(regr.cv[[fold]], newdata=data,
+                                                type="response")[test.idx])
+   }
+   return(pred.cv)
+ }
```

Report the mean cross-validated AUCs:

```
> pred.cv.m1 <- predict.cv(cv.m1, chdage, chdage$CHD, folds)
> pred.cv.m2 <- predict.cv(cv.m2, chdage, chdage$CHD, folds)
> auc.cv.m1 <- auc.cv.m2 <- numeric(length(folds))
> for (fold in 1:length(folds)) {
+   auc.cv.m1[fold] <- roc(obs ~ pred, data=pred.cv.m1[[fold]])$auc
+   auc.cv.m2[fold] <- roc(obs ~ pred, data=pred.cv.m2[[fold]])$auc
+ }
> round(mean(auc.cv.m1), 3)
[1] 0.756
> round(mean(auc.cv.m2), 3)
[1] 0.754
```

Question 2

Convert the group variable to 0-1:

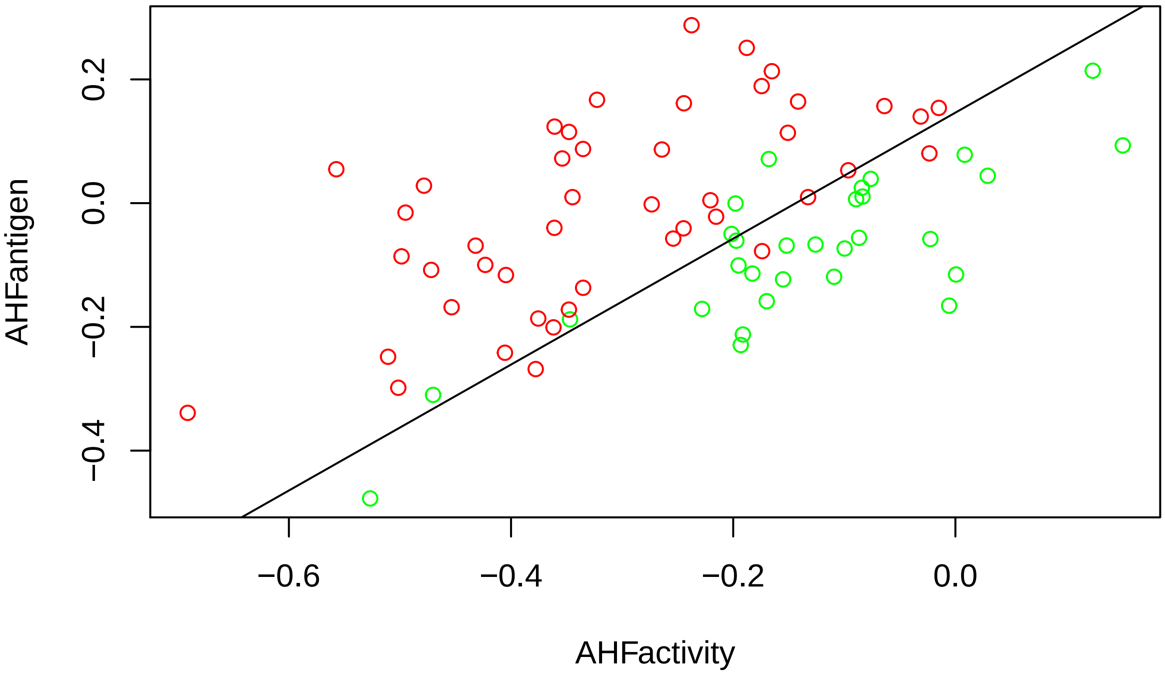
```
> hemo <- read.csv("data/hemophilia.csv")
> hemo$group <- 2 - as.integer(hemo$group)  # assign a 1 to carriers
```

Fit a logistic regression model and retrieve the probabilities:

```
> regr <- glm(group ~ AHFantigen + AHFactivity, data=hemo, family="binomial")
>
> # option 1
> probs <- regr$fitted.values
>
> # option 2
> probs <- exp(regr$linear.predictors) / (exp(regr$linear.predictors) + 1)
```

Scatter plot of the data points and decision boundary:

```
> group.col <- hemo$group
> group.col[group.col == 0] <- "green"
> group.col[group.col == 1] <- "red"
> with(hemo, plot(AHFactivity, AHFantigen, col=group.col))
> intercept <- -coef(regr)[1]/(coef(regr)[2])
> slope <- -coef(regr)[3]/(coef(regr)[2])
> abline(intercept, slope)
```



Count number of misclassified observations for $\theta = 0.5$:

```
> pred.case <- as.integer(probs > 0.5)
> sum(pred.case != hemo$group)
[1] 9
```

Derive sensitivity and specificity:

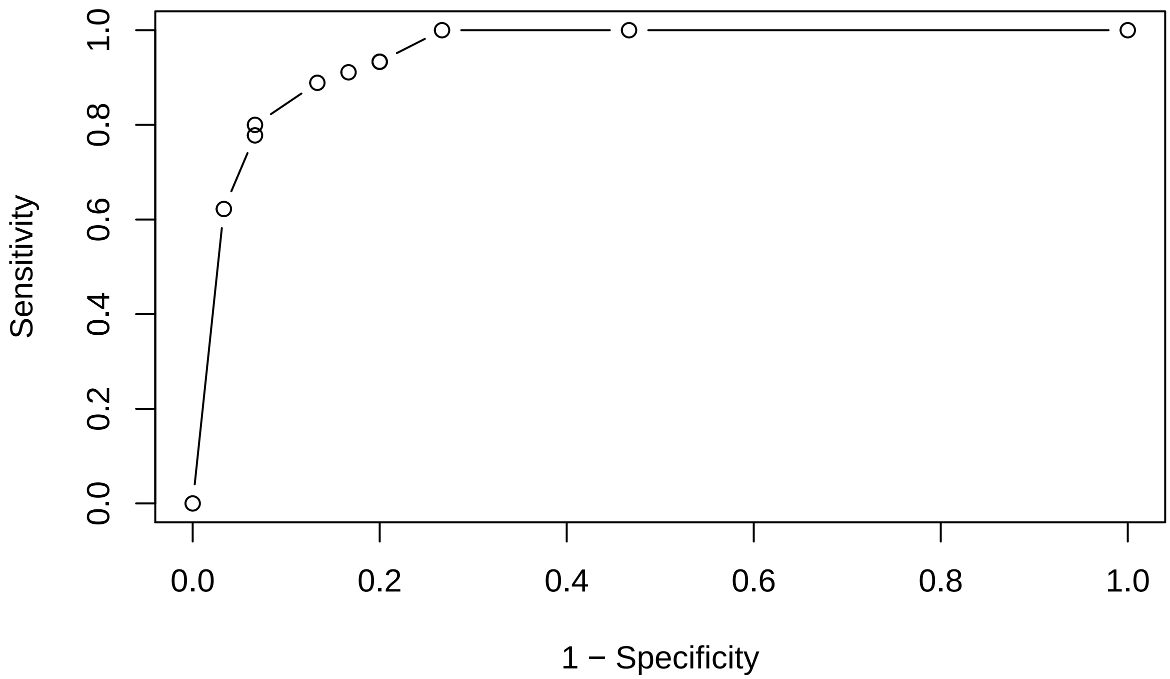
```
> sens <- sum(hemo$group == 1 & pred.case == 1) / sum(hemo$group == 1)
> round(sens, 2)
[1] 0.91
> spec <- sum(hemo$group == 0 & pred.case == 0) / sum(hemo$group == 0)
> round(spec, 2)
[1] 0.83
```

Function `sens.spec()`:

```
> sens.spec <- function(y.obs, y.pred, threshold) {
+   sens <- sum(y.obs == 1 & (y.pred > threshold) == 1) / sum(y.obs == 1)
+   spec <- sum(y.obs == 0 & (y.pred > threshold) == 0) / sum(y.obs == 0)
+   return(c(sens, spec))
+ }
```

Compute sensitivity and specificity for values of θ between 0 and 1 and plot them in the ROC space:

```
> ss <- NULL
> for (thresh in seq(0, 1, by=0.1)) {
+   ss <- rbind(ss, sens.spec(hemo$group, probs, thresh))
+ }
> plot(1 - ss[, 2], ss[, 1], xlab="1 - Specificity", ylab="Sensitivity",
+      type="b")              # plots points and joining them with lines
```



Question 3

Compute the odds ratio for exposure to smoking in parents:

```
> round((816/188) / (3203/1168), 2)
[1] 1.58
```

Create a synthetic dataset and fit a logistic regression model:

```
> par.smoke <- c(rep(1, 816 + 3203), rep(0, 188 + 1168))
> stu.smoke <- c(rep(1, 816), rep(0, 3203), rep(1, 188), rep(0, 1168))
> regr.smoke <- glm(stu.smoke ~ par.smoke, family="binomial")
```

Odds ratio, 95% confidence interval and p-value:

```
> round(exp(coef(regr.smoke)[2]), 2)
par.smoke
     1.58
> round(exp(confint(regr.smoke)[2, ]), 2)
 2.5 % 97.5 %
  1.34   1.88
> signif(coef(summary(regr.smoke))[2, 4], 3)
[1] 1.71e-07
```

Test of goodness-of-fit:

```
> signif(pchisq(regr.smoke$null.deviance - regr.smoke$deviance, df=1,
+               lower.tail=FALSE), 2)
[1] 6.8e-08
```

Probability of smoking for a student whose parents are not smokers:

```
> # option 1
> prob.par.smoke0 <- exp(coef(regr.smoke)[1]) / (1 + exp(coef(regr.smoke)[1]))
> round(prob.par.smoke0, 3)
(Intercept)
      0.139
>
> # option 2
> predict(regr.smoke, newdata=data.frame(par.smoke=0), type="response")
        1
0.1386431
```

Probability of smoking for a student whose parents are smokers:

```
> # option 1
> prob.par.smoke1 <- exp(sum(coef(regr.smoke))) / (1 + exp(sum(coef(regr.smoke))))
> round(prob.par.smoke1, 3)
[1] 0.203
>
> # option 2
> predict(regr.smoke, newdata=data.frame(par.smoke=1), type="response")
        1
0.2030356
```

Sensitivity and specificity for an appropriate θ (any value between 0.139 and 0.203 is fine as in this model there are only two predicted probabilities):

```
> sens <- sum(fitted(regr.smoke) > 0.15 & stu.smoke) / sum(stu.smoke)
> round(sens, 3)
[1] 0.813
> spec <- sum(fitted(regr.smoke) < 0.15 & (1 - stu.smoke)) / sum(1 - stu.smoke)
> round(spec, 3)
[1] 0.267
```