5.1.1 - R: Logistic Regression

Stat 5100: Dr. Bean

Example: (Text Table 14.3) Individuals were randomly sampled within two sectors of a city, and checked for presence of disease (here, spread by mosquitoes). Subjects' age (in years), socioeconomic status (low, medium, high), and city sector are to be used to predict the probability of contracting the disease.

In R, we can create logistic regression models with the glm function. "GLM" stands for generalized linear model, and can be used to fit a variety of linear models. To specify logistic regression, we set an option in side the glm function that specifies a binomial (two classes) response.

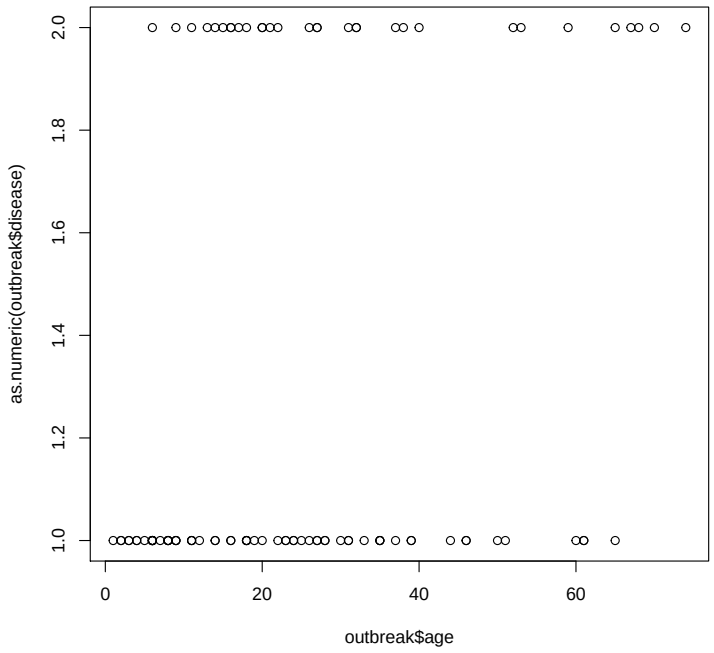
Fit a logistic regression model

```
# Input the data
library(stat5100)
data(outbreak)

outbreak_logreg <- glm(disease ~ age + SES_mid + SES_low + sector,
                       data = outbreak, family = "binomial")
summary(outbreak_logreg)

##
## Call:
## glm(formula = disease ~ age + SES_mid + SES_low + sector, family = "binomial",
##     data = outbreak)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6552  -0.7529  -0.4788   0.8558   2.0977
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.31293    0.64259  -3.599 0.000319 ***
## age          0.02975    0.01350   2.203 0.027577 *
## SES_mid1     0.40879    0.59900   0.682 0.494954
## SES_low1    -0.30525    0.60413  -0.505 0.613362
## sector1      1.57475    0.50162   3.139 0.001693 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122.32  on 97  degrees of freedom
## Residual deviance: 101.05  on 93  degrees of freedom
## AIC: 111.05
##
## Number of Fisher Scoring iterations: 4
```

Plot a graph of observed values and predicted probabilities

Because we have multiple predictor variables, we have to choose one variable to put on the x-axis, and the rest of the variables will have to be fixed at some value. We will use Age for our x-axis variable. We will set $\text{SES}_m id = 0.245, SES_l ow = 0.367, and Sector = 0.398$.

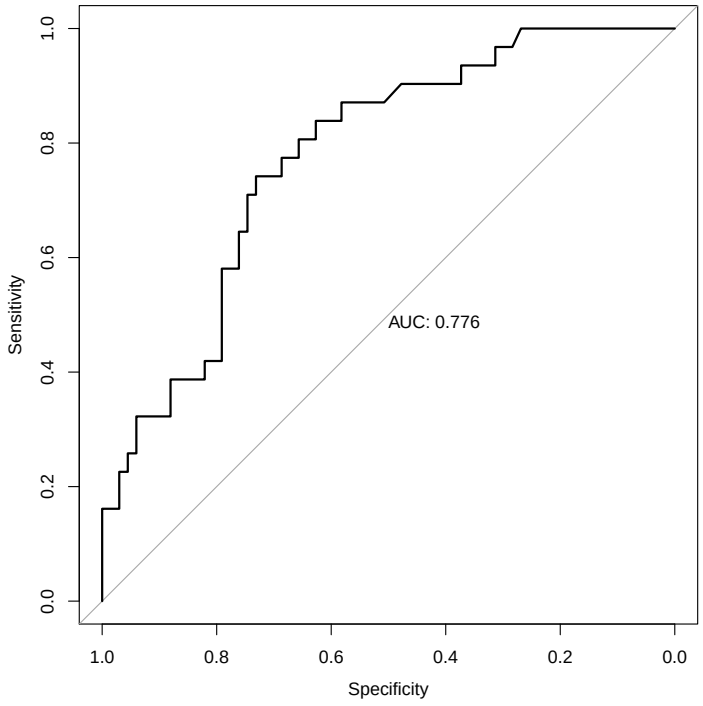
```
# Get a range of ages, and then predict the probability with the predict()
# function to get the shape of the predicted probability curve.
age_range <- seq(0, 80, length = 500)
npred <- length(age_range)

pred_data <- data.frame(age = age_range, SES_mid = as.factor(rep(0.245, npred)),
                        SES_low = as.factor(rep(0.367, npred)),
                        sector = as.factor(rep(0.398, npred)))
pred_disease <- predict(outbreak_logreg, newdata = pred_data, type = "response")

## Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels):
factor SES_mid has new level 0.245

plot(outbreak$age, as.numeric(outbreak$disease))
```




```
# Input the data
library(stat5100)
data(outbreak)

outbreak_logreg <- glm(disease ~ age + SES_mid + SES_low + sector,
                       data = outbreak, family = "binomial")
summary(outbreak_logreg)

##
## Call:
## glm(formula = disease ~ age + SES_mid + SES_low + sector, family = "binomial",
##     data = outbreak)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6552  -0.7529  -0.4788   0.8558   2.0977
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.31293    0.64259  -3.599 0.000319 ***
## age          0.02975    0.01350   2.203 0.027577 *
## SES_mid1     0.40879    0.59900   0.682 0.494954
## SES_low1    -0.30525    0.60413  -0.505 0.613362
## sector1      1.57475    0.50162   3.139 0.001693 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122.32  on 97  degrees of freedom
## Residual deviance: 101.05  on 93  degrees of freedom
## AIC: 111.05
##
## Number of Fisher Scoring iterations: 4
```

```
# ROC Curve
prob <- predict(outbreak_logreg, newdata = outbreak, type = "response")
pROC::roc(outbreak$disease ~ prob, plot = TRUE, print.auc = TRUE)

## Setting levels:  control = 0, case = 1
## Setting direction:  controls < cases
```



```
##
## Call:
## roc.formula(formula = outbreak$disease ~ prob, plot = TRUE, print.auc = TRUE)
##
## Data: prob in 67 controls (outbreak$disease 0) < 31 cases (outbreak$disease 1).
## Area under the curve: 0.7764
```