Answers to exercises Session 5

Marjolein Fokkema

Exercise 1

Read in data:

```
load("MASQ.Rda")
set.seed(1)
train <- sample(1:nrow(MASQ), size = nrow(MASQ)*.8)
summary(MASQ)
```

```
##  D_DEPDYS       AD               AA              GDD             GDA
##  0:1927   Min.   : 26.00   Min.   :17.00   Min.   :12.00   Min.   :11.0
##  1:1670   1st Qu.: 64.00   1st Qu.:22.00   1st Qu.:20.00   1st Qu.:19.0
##           Median : 77.00   Median :28.00   Median :29.00   Median :24.0
##           Mean   : 75.05   Mean   :32.01   Mean   :30.64   Mean   :25.4
##           3rd Qu.: 88.00   3rd Qu.:39.00   3rd Qu.:40.00   3rd Qu.:31.0
##           Max.   :110.00   Max.   :83.00   Max.   :60.00   Max.   :54.0
##       GDM          leeftijd    geslacht
##  Min.   :15.0   Min.   :17.0   m:1317
##  1st Qu.:31.0   1st Qu.:28.0   v:2280
##  Median :40.0   Median :38.0
##  Mean   :40.6   Mean   :38.8
##  3rd Qu.:50.0   3rd Qu.:48.0
##  Max.   :75.0   Max.   :91.0
```

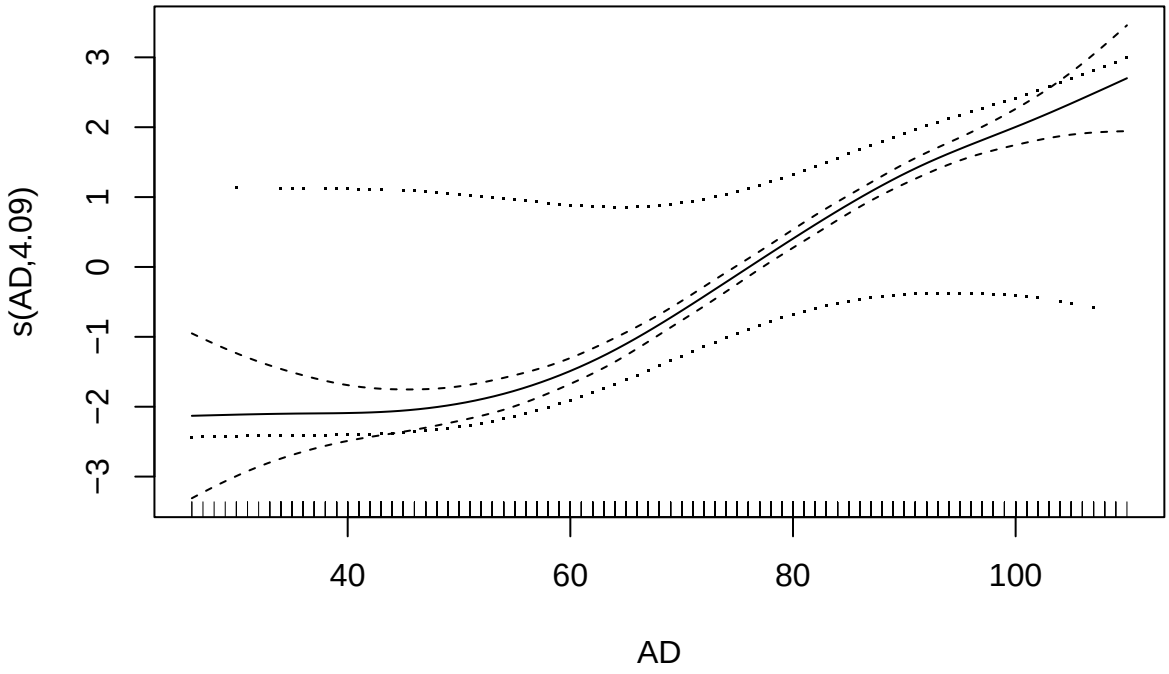
Fit a smoothing spline of the `AD` variable to predict `D_DEPDYS`:

```
library("mgcv")
GAM <- gam(D_DEPDYS ~ s(AD, bs = "cr"),
           data = MASQ[train, ], method = "REML", family = "binomial")
summary(GAM)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## D_DEPDYS ~ s(AD, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.24089    0.04751   -5.07 3.98e-07 ***
## ---
```

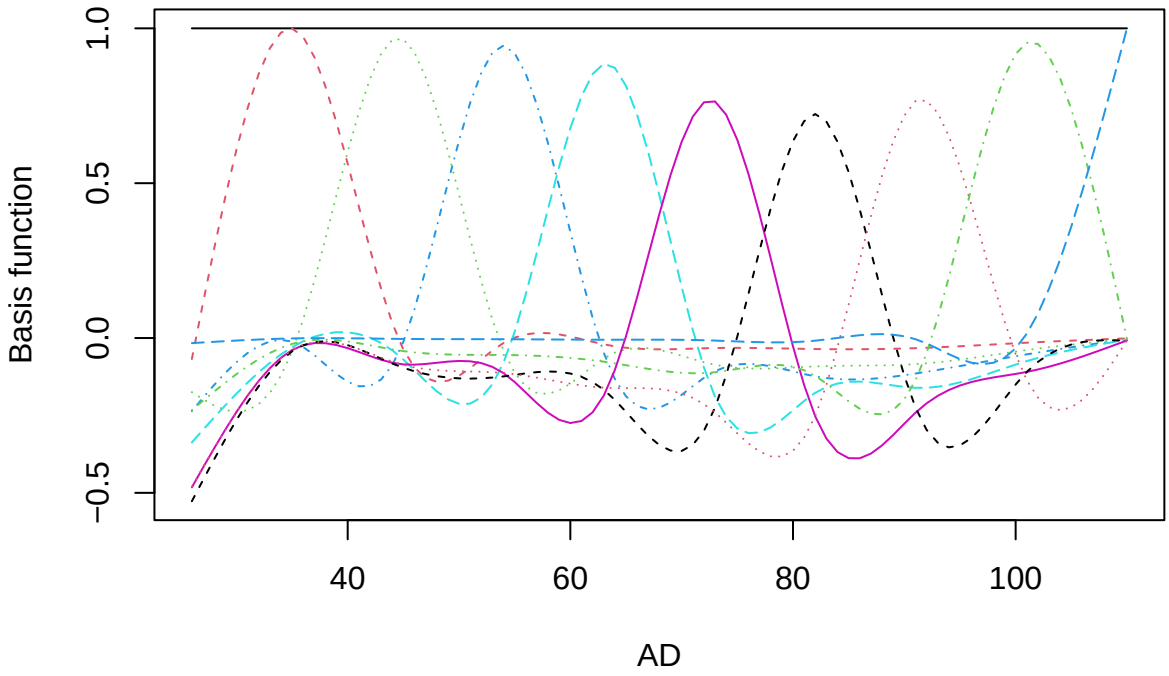
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##        edf Ref.df Chi.sq p-value
## s(AD) 4.09  5.041  672.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.297   Deviance explained = 23.7%
## -REML = 1522.2  Scale est. = 1         n = 2877
```

```
plot(GAM, residuals = TRUE)
```



Inspect the basis functions that were created for `AD`:

```
mod_mat <- model.matrix(GAM)
matplot(sort(MASQ$AD[train]), mod_mat[order(MASQ$AD[train]), ], type = "l",
        xlab = "AD", ylab = "Basis function")
```



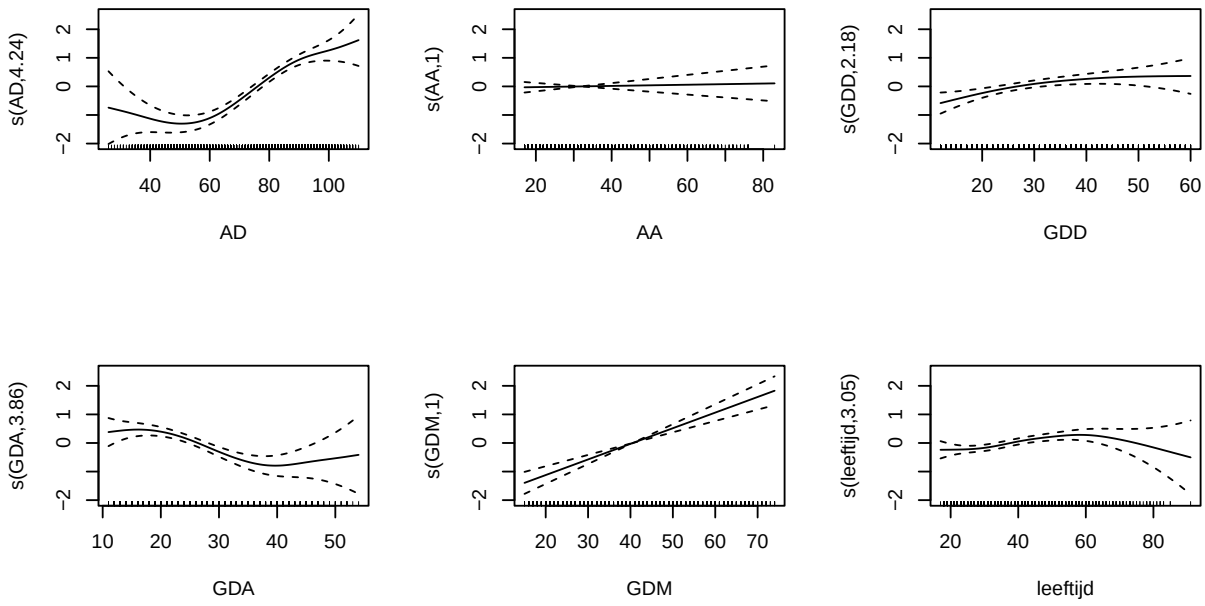
Exercise 2: Multiple predictors

```
library("mgcv")
GAM <- gam(D_DEPDYS ~ s(AD) + s(AA) + s(GDD) + s(GDA) + s(GDM) + s(leeftijd) + geslacht,
           data = MASQ[train, ], method = "REML", family = "binomial")
summary(GAM)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## D_DEPDYS ~ s(AD) + s(AA) + s(GDD) + s(GDA) + s(GDM) + s(leeftijd) +
##     geslacht
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.33822    0.07755  -4.361 1.29e-05 ***
## geslachtv    0.13314    0.09517   1.399    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
```

```
##                edf Ref.df  Chi.sq  p-value
## s(AD)        4.243  5.238 148.910  < 2e-16 ***
## s(AA)        1.001  1.002   0.115  0.73591
## s(GDD)       2.185  2.793  11.185  0.00971 **
## s(GDA)       3.855  4.806  33.279 4.62e-06 ***
## s(GDM)       1.001  1.001  52.723  < 2e-16 ***
## s(leeftijd) 3.050  3.817  17.530  0.00137 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.331   Deviance explained =   27%
## -REML = 1478.1  Scale est. = 1         n = 2877
```

```
par(mfrow = c(2, 3))
plot(GAM)
```



We compute the mean squared error and misclassification rate using predicted probabilities, for both training and test observations:

```
y_train <- as.numeric(MASQ[train, "D_DEPDYS"]) - 1
y_test <- as.numeric(MASQ[-train, "D_DEPDYS"]) - 1

## Training data
GAM_preds_train <- predict(GAM, newdata = MASQ[train, ], type = "response")
mean((y_train - GAM_preds_train)^2) ## Brier score
```

```
## [1] 0.1653101
```

```r
tab_train <- prop.table(table(MASQ[train, "D_DEPDYS"], GAM_preds_train > .5)) ## confusion matrix
tab_train
```

```
##
##         FALSE      TRUE
##   0 0.4174487 0.1223497
##   1 0.1160932 0.3441084
```

```r
1 - sum(diag(tab_train)) ## MCR
```

```
## [1] 0.2384428
```

```r
## Test data
GAM_preds_test <- predict(GAM, newdata = MASQ[-train, ], type = "response")
mean((y_test - GAM_preds_test)^2) ## Brier score
```

```
## [1] 0.1666467
```

```r
tab_test <- prop.table(table(MASQ[-train, "D_DEPDYS"], GAM_preds_test > .5)) ## confusion matrix
tab_test
```

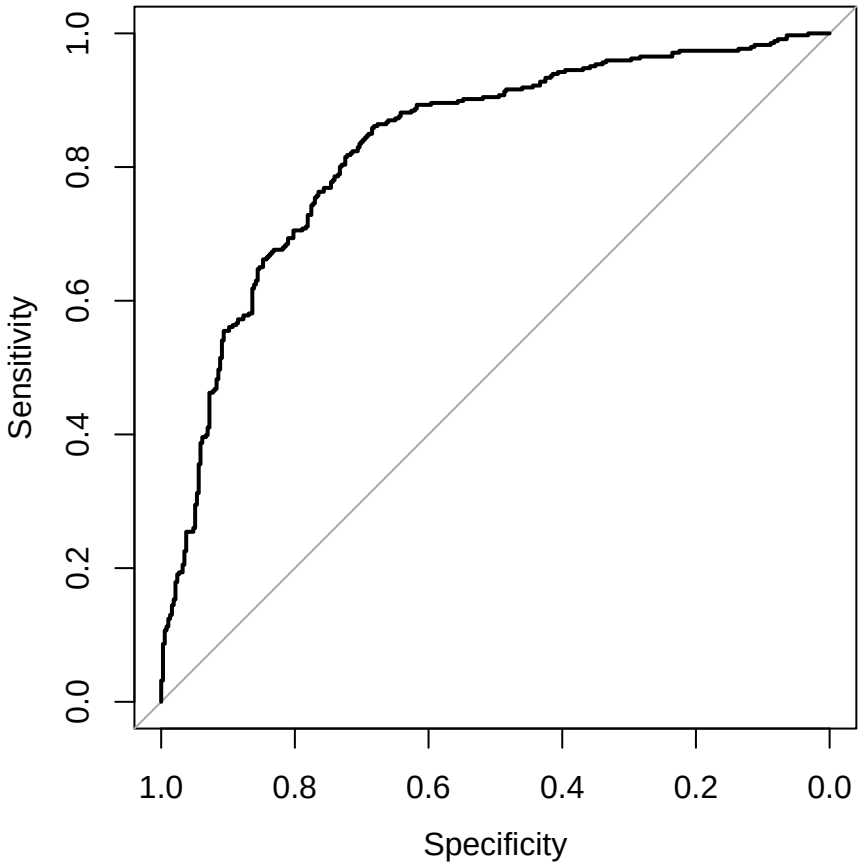
```
##
##         FALSE      TRUE
##   0 0.4027778 0.1166667
##   1 0.1236111 0.3569444
```

```r
1 - sum(diag(tab_test)) ## MCR
```

```
## [1] 0.2402778
```

The Brier score and confusion matrices are quite similar between training and test data, indicating little overfitting.

```r
## Or, compute ROC curve
library("pROC")
plot(roc(resp = y_test, pred = GAM_preds_test))
```

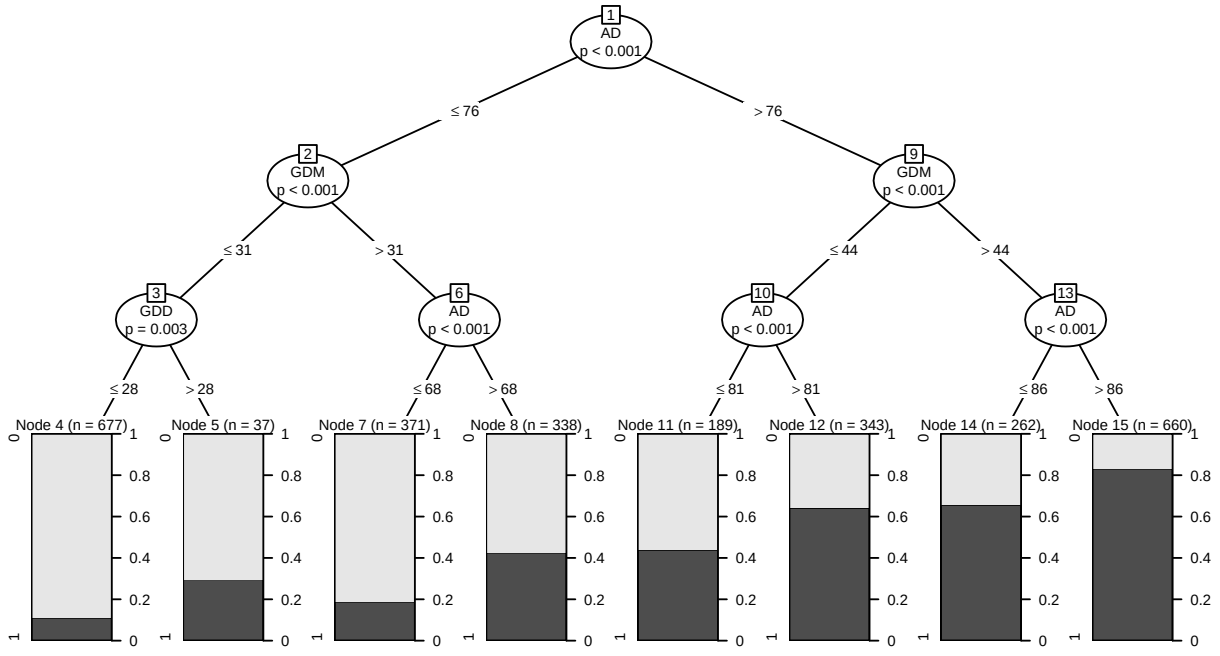



```
auc(resp = y_test, pred = GAM_preds_test)
```

```
## Area under the curve: 0.8295
```

Exercise 4: Fit a ctree to MASQ data

```
library("partykit")
ct <- ctree(D_DEPDYS ~ . , data = MASQ[train, ])
plot(ct, gp = gpar(cex = .5))
```



The conditional inference tree indicates a positive effect of the AD, GDM and GDD subscales on the probability of having a depressive / dysthymic disorder.

```
## Training data
ct_preds_train <- predict(ct, newdata = MASQ[train, ], type = "prob")[ , 2]
mean((y_train - ct_preds_train)^2) ## Brier score
```

```
## [1] 0.1705674
```

```
tab_train <- prop.table(table(MASQ[train, "D_DEPDYS"], ct_preds_train > .5)) ## confusion matrix
1 - sum(diag(tab_train)) ## MCR
```

```
## [1] 0.2457421
```

```
## Test data
y_test <- as.numeric(MASQ[-train, "D_DEPDYS"]) - 1
ct_preds_test <- predict(ct, newdata = MASQ[-train, ], type = "prob")[ , 2]
mean((y_test - ct_preds_test)^2) ## Brier score
```

```
## [1] 0.1738697
```

```r
tab_test <- prop.table(table(MASQ[-train, "D_DEPDYS"], ct_preds_test > .5)) ## confusion matrix
1 - sum(diag(tab_test)) ## MCR
```

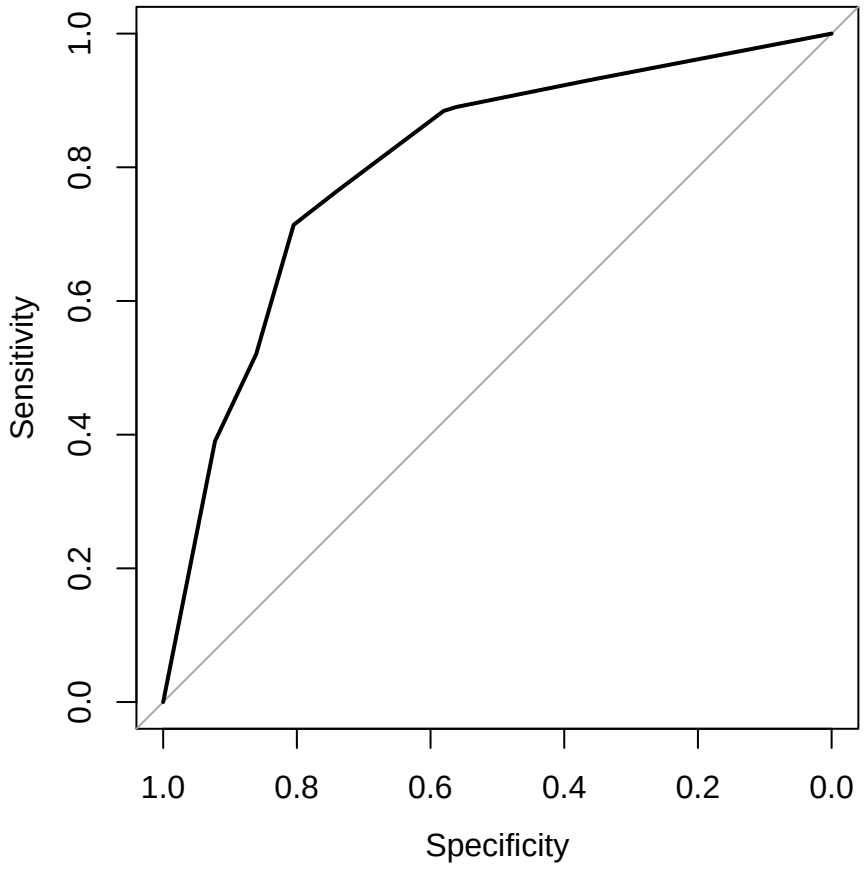
```
## [1] 0.2388889
```

The conditional inference tree provided best predictive accuracy of the single trees.

```r
## AUC on test observations
plot(roc(resp = y_test, pred = ct_preds_test))
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```




```r
auc(resp = y_test, pred = ct_preds_test)
```

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

## Area under the curve: 0.805
```