Assignment 2

Biomedical Data Science (MATH11174), 22/23, Semester 2

Aysu Ismayilova

April 6, 2023

Due on Thursday, 6th of April 2023, 5:00pm

! Pay Attention

The assignment is marked out of 100 points, and will contribute to ***30%*** of your final mark. The aim of this assignment is to produce a precise report in biomedical studies with the help of statistical and machine learning. Please complete this assignment using **Quarto/Rmarkdown file and render/knit this document only in PDF format** (rendering while solving the questions will prevent sudden panic before submission!). Submit using the **gradescope link on Learn** and ensure that **all questions are tagged accordingly**. You can simply click render on the top left of Rstudio (`Ctrl+Shift+K`). If you cannot render/knit to PDF directly, open **Terminal** in your RStudio (`Alt+Shift+R`) and type `quarto tools install tinytex`, otherwise please follow this link. If you have any code that does not run you will not be able to render nor knit the document so comment it as you might still get some grades for partial code.

Codes that are **clear and reusable will be rewarded**. Codes without proper indentation, choice of variable identifiers, **comments**, efficient code, etc will be penalised. An initial code chunk is provided after each subquestion but **create as many chunks as you feel is necessary** to make a clear report. Add plain text explanations in between the chunks when required to make it easier to follow your code and reasoning. Ensure that all answers containing multiple values should be presented and formatted only with `kable()` and `kable_styling()` otherwise penalised (**no use of `print()` or `cat()`**). All plots must be displayed with clear title, label and legend otherwise penalised.

This is an **individual assignment**, and **no public discussions** will be allowed. If you have any question, please ask on Piazza by specifying your `Post to` option to `instructors`. To join Piazza, please follow this link.

Problem 1 (27 points)

File `wdbc2.csv` (available from the accompanying zip folder on Learn) refers to a study of breast cancer where the outcome of interest is the type of the tumour (benign or malignant, recorded in column `diagnosis`). The study collected 30 imaging biomarkers on 569 patients.

Problem 1.a (7 points)

- Using package `caret`, create a data partition so that the training set contains 70% of the observations (set the random seed to 984065 beforehand).
- Fit both a ridge and Lasso regression model which use cross validation on the training set to diagnose the type of tumour from the 30 biomarkers.
- Then use a plot to help identify the penalty parameter λ that maximises the AUC and report the λ for both ridge and Lasso regression using `kable()`.
- ***Note : there is no need to use the `prepare.glmnet()` function from lab 4, using `as.matrix()` with the required columns is sufficient.***

```
# Load required packages
library(caret)
library(glmnet)
library(pROC)
library(knitr)

# Load data
wdbc2 <- read.csv("data_assignment2/wdbc2.csv", stringsAsFactors = F)
wdbc2 <- na.omit(wdbc2)
wdbc2$diagnosis <- ifelse(wdbc2$diagnosis == "malignant", 1, 0)

# Set random seed for reproducibility
set.seed(984065)




folds <- createFolds(wdbc2$diagnosis, k=10)

## function from Lab 3
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)
}

lr.cv <- glm.cv(diagnosis ~ .,
                data=wdbc2, folds)


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)
}
pred.lr.cv <- predict.cv(lr.cv, wdbc2, wdbc2$diagnosis, folds)
```

```
auc.lr.cv <- numeric(length(folds))
suppressMessages(invisible({
  for (fold in 1:length(folds)) {
    auc.lr.cv[fold] <- roc(obs ~ pred, data = pred.lr.cv[[fold]])$auc
  }
}))
round(mean(auc.lr.cv), 3)
```

```
[1] 0.956
```

```
y.wdbc2 <- as.matrix(wdbc2$diagnosis)
x.wdbc2 <- as.matrix(wdbc2[, 3:32])
ridge.cv <- pred.ridge.cv <- NULL
for (fold in 1:length(folds)) {
  test.idx <- folds[[fold]]
  ridge.cv[[fold]] <-
    cv.glmnet(x.wdbc2[-test.idx,], y.wdbc2[-test.idx],
              family = "binomial", alpha = 0)
  lambda.min <- ridge.cv[[fold]]$lambda.min
```

```r
  pred.ridge.cv[[fold]] <- data.frame(
    obs = y.wdbc2[test.idx],
    pred = predict(
      ridge.cv[[fold]],
      newx = x.wdbc2[test.idx,],
      type = "response",
      s = lambda.min
    )[, 1]
  )
}
auc.ridge.cv <- numeric(length(folds))

suppressMessages(invisible({
  for (fold in 1:length(folds)) {
    auc.ridge.cv[fold] <- roc(obs ~ pred,
                              data = pred.ridge.cv[[fold]])$auc
  }
}))




lr.coefs <- coef(lr.cv[[1]])[-1] # ignore the intercept
lr.coefs <- lr.coefs [-1]
lambda.idx <-
  which(ridge.cv[[1]]$lambda == ridge.cv[[1]]$lambda.min)
ridge.coefs <- ridge.cv[[1]]$glmnet.fit$beta[, lambda.idx]
df <-
  round(data.frame(lr.coefs, ridge.coefs, ratio = ridge.coefs / lr.coefs),
        3)

kable(df, caption = "Lasso vs Ridge") |>
  kable_styling(full_width = F,
                position = "center",
                latex_options = "hold_position")
```

Table 1: Lasso vs Ridge

	lr.coefs	ridge.coefs	ratio
radius	0.270	0.128	0.475
texture	-0.018	0.075	-4.260
perimeter	0.004	0.011	2.550
area	-0.017	0.000	-0.011
smoothness	-20.938	14.311	-0.683
compactness	-18.761	-0.163	0.009
concavity	-73.518	2.025	-0.028
concavepoints	199.098	6.749	0.034
symmetry	-5.462	1.102	-0.202
fractaldimension	-82.268	-23.765	0.289
radius.stderr	2.992	1.387	0.464
texture.stderr	-1.871	-0.077	0.041
perimeter.stderr	0.424	0.116	0.273
area.stderr	0.075	0.006	0.077
smoothness.stderr	124.844	-17.740	-0.142
compactness.stderr	-156.830	-12.247	0.078
concavity.stderr	54.785	3.919	0.072
concavepoints.stderr	24.437	22.830	0.934
symmetry.stderr	109.067	-12.959	-0.119
fractaldimension.stderr	174.304	-34.845	-0.200
radius.worst	5.583	0.068	0.012
texture.worst	0.395	0.064	0.162
perimeter.worst	-0.091	0.008	-0.088
area.worst	-0.035	0.000	-0.008
smoothness.worst	40.046	14.236	0.355
compactness.worst	8.077	0.438	0.054
concavity.worst	19.664	1.128	0.057
concavepoints.worst	-28.112	5.254	-0.187
symmetry.worst	2.171	5.210	2.400
fractaldimension.worst	-0.225	5.519	-24.498

Problem 1.b (2 points)

- Create a data table that for each value of `lambda.min` and `lambda.1se` for each model fitted in **problem 1.a** that contains the corresponding λ, AUC and model size.
- Use 3 significant figures for floating point values and comment on these results.
- ***Note : The AUC values are stored in the field called*** `cvm`.

```r
# Fit Lasso model with cross-validation
lasso.cv <- pred.lasso.cv <- NULL
for (fold in 1:length(folds)) {
  test.idx <- folds[[fold]]
  lasso.cv[[fold]] <-
    cv.glmnet(x.wdbc2[-test.idx,], y.wdbc2[-test.idx],
              family = "binomial", alpha = 1)
  lambda.min <- lasso.cv[[fold]]$lambda.min
  pred.lasso.cv[[fold]] <- data.frame(
    obs = y.wdbc2[test.idx],
    pred = predict(
      lasso.cv[[fold]],
      newx = x.wdbc2[test.idx,],
      type = "response",
      s = lambda.min
    )[, 1]
  )
}
auc.lasso.cv <- numeric(length(folds))
suppressMessages(invisible({
  for (fold in 1:length(folds)) {
    auc.lasso.cv[fold] <- roc(obs ~ pred,
                              data = pred.lasso.cv[[fold]])$auc
  }
}))
```

```r
# Extract lambda, AUC, and model size for lambda.min and lambda.1se
lambda <- c(lasso.cv[[1]]$lambda.min, lasso.cv[[1]]$lambda.1se)
auc <-
  c(lasso.cv[[1]]$cvm[lasso.cv[[1]]$lambda == lasso.cv[[1]]$lambda.min],
    lasso.cv[[1]]$cvm[lasso.cv[[1]]$lambda == lasso.cv[[1]]$lambda.1se])
model_size <-
  c(sum(coef(lasso.cv[[1]], s = lasso.cv[[1]]$lambda.min) != 0),
    sum(coef(lasso.cv[[1]], s = lasso.cv[[1]]$lambda.1se) != 0))

# Create data table
results <-
  data.frame(lambda = lambda,
             auc = auc,
             model_size = model_size)
row.names(results) <- c("lambda.min", "lambda.1se")
```

```r
results <- round(results, 3)

# Fit Ridge model with cross-validation
ridge.cv <-
  cv.glmnet(x.wdbc2, y.wdbc2, family = "binomial", alpha = 0)

# Extract lambda, AUC, and model size for lambda.min and lambda.1se
lambda_2 <- c(ridge.cv$lambda.min, ridge.cv$lambda.1se)
auc <- c(ridge.cv$cvm[ridge.cv$lambda == ridge.cv$lambda.min],
         ridge.cv$cvm[ridge.cv$lambda == ridge.cv$lambda.1se])
model_size <- c(sum(coef(ridge.cv, s = ridge.cv$lambda.min) != 0),
                sum(coef(ridge.cv, s = ridge.cv$lambda.1se) != 0))

# Create data table
results_ridge <-
  data.frame(lambda = lambda_2,
             auc = auc,
             model_size = model_size)
row.names(results_ridge) <- c("lambda.min", "lambda.1se")
results_ridge <- round(results_ridge, 3)


# Create data frames for Lasso and Ridge results
results <- data.frame(
  Model = "Lasso",
  Lambda = lambda,
  Lambda_Min = ifelse(lambda == lasso.cv[[1]]$lambda.min, "Yes", ""),
  Lambda_1SE = ifelse(lambda == lasso.cv[[1]]$lambda.1se, "Yes", ""),
  AUC = auc,
  Model_Size = model_size
)
results_ridge <- data.frame(
  Model = "Ridge",
  Lambda = lambda_2,
  Lambda_Min = ifelse(lambda_2 == ridge.cv$lambda.min, "Yes", ""),
  Lambda_1SE = ifelse(lambda_2 == ridge.cv$lambda.1se, "Yes", ""),
  AUC = auc,
  Model_Size = model_size
)

# Combine results of Lasso and Ridge models
```

```r
combined_results <- rbind(results, results_ridge)

# Print combined table
kable(combined_results, caption = "Data table for lambda.min and lambda.1se for the Lasso
  kable_styling(full_width = F,
                position = "center",
                latex_options = "hold_position")
```

Table 2: Data table for lambda.min and lambda.1se for the Lasso and Ridge models fitted in problem 1.a that contains the corresponding model, lambda, AUC, and model size, along with information about which lambda value corresponds to lambda.min and lambda.1se

Model	Lambda	Lambda_Min	Lambda_1SE	AUC	Model_Size
Lasso	0.0066585	Yes		0.3951141	31
Lasso	0.0323777		Yes	0.4576177	31
Ridge	0.0367934	Yes		0.3951141	31
Ridge	0.1963552		Yes	0.4576177	31

Problem 1.c (7 points)

- Perform both backward (we denote this as **model B**) and forward (**model S**) stepwise selection on the same training set derived in **problem 1.a**. Mute all the trace by setting `trace = FALSE`.
- Report the variables selected and their standardised regression coefficients in increasing order of the absolute value of their standardised regression coefficient.
- Discuss the results and how the different variables entering or leaving the model influenced the final result.
- ***Note : You can mute the warning by assigning {r warning = FALSE} for the chunk title***

```r
# Fit the full model

full.model <- glm(diagnosis ~ ., data = wdbc2, family = "binomial")

# Perform backward stepwise selection
sel.back <-
  stepAIC(full.model, direction = "back") # backward elimination
```

```
Start:  AIC=182.66
```

```
diagnosis ~ id + radius + texture + perimeter + area + smoothness +
    compactness + concavity + concavepoints + symmetry + fractaldimension +
    radius.stderr + texture.stderr + perimeter.stderr + area.stderr +
    smoothness.stderr + compactness.stderr + concavity.stderr +
    concavepoints.stderr + symmetry.stderr + fractaldimension.stderr +
    radius.worst + texture.worst + perimeter.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                         Df Deviance    AIC
- symmetry                1   118.66 180.66
- compactness             1   118.67 180.67
- texture                 1   118.69 180.69
- perimeter.stderr        1   118.79 180.79
- smoothness.stderr       1   118.80 180.80
- concavepoints.stderr    1   118.85 180.85
- area.stderr             1   118.86 180.86
- perimeter               1   118.88 180.88
- concavepoints.worst     1   118.90 180.90
- symmetry.worst          1   118.92 180.92
- smoothness              1   119.08 181.08
- fractaldimension        1   119.13 181.13
- fractaldimension.stderr 1   119.17 181.18
- symmetry.stderr         1   119.19 181.19
- perimeter.worst         1   119.34 181.34
- fractaldimension.worst  1   119.50 181.50
- compactness.worst       1   119.67 181.67
- compactness.stderr      1   119.72 181.72
- id                      1   119.81 181.81
- smoothness.worst        1   120.27 182.26
- texture.stderr          1   120.52 182.52
<none>                        118.66 182.66
- radius.stderr           1   121.35 183.35
- radius                  1   121.69 183.69
- radius.worst            1   122.04 184.04
- concavity.worst         1   122.98 184.98
- texture.worst           1   123.83 185.84
- concavity               1   123.91 185.91
- area                    1   124.61 186.61
- concavity.stderr        1   125.97 187.97
- concavepoints           1   126.83 188.83
- area.worst              1   131.00 193.00
```

```
Step:  AIC=180.66
diagnosis ~ id + radius + texture + perimeter + area + smoothness +
    compactness + concavity + concavepoints + fractaldimension +
    radius.stderr + texture.stderr + perimeter.stderr + area.stderr +
    smoothness.stderr + compactness.stderr + concavity.stderr +
    concavepoints.stderr + symmetry.stderr + fractaldimension.stderr +
    radius.worst + texture.worst + perimeter.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                         Df Deviance    AIC
- compactness             1   118.67 178.67
- texture                 1   118.69 178.69
- perimeter.stderr        1   118.80 178.79
- smoothness.stderr       1   118.80 178.80
- concavepoints.stderr    1   118.85 178.85
- area.stderr             1   118.86 178.86
- perimeter               1   118.88 178.88
- concavepoints.worst     1   118.92 178.92
- symmetry.worst          1   119.01 179.01
- smoothness              1   119.10 179.10
- fractaldimension        1   119.16 179.16
- fractaldimension.stderr 1   119.18 179.18
- symmetry.stderr         1   119.21 179.21
- perimeter.worst         1   119.35 179.35
- fractaldimension.worst  1   119.51 179.51
- compactness.worst       1   119.70 179.70
- compactness.stderr      1   119.73 179.73
- id                      1   119.81 179.81
- smoothness.worst        1   120.29 180.29
<none>                        118.66 180.66
- texture.stderr          1   120.69 180.69
- radius.stderr           1   121.35 181.35
- radius                  1   121.69 181.69
- radius.worst            1   122.05 182.05
- concavity.worst         1   122.99 182.99
- texture.worst           1   123.94 183.94
- concavity               1   123.97 183.97
- area                    1   124.72 184.72
- concavity.stderr        1   126.02 186.02
- concavepoints           1   126.98 186.99
- area.worst              1   131.01 191.01
```

```
Step:  AIC=178.67
diagnosis ~ id + radius + texture + perimeter + area + smoothness +
    concavity + concavepoints + fractaldimension + radius.stderr +
    texture.stderr + perimeter.stderr + area.stderr + smoothness.stderr +
    compactness.stderr + concavity.stderr + concavepoints.stderr +
    symmetry.stderr + fractaldimension.stderr + radius.worst +
    texture.worst + perimeter.worst + area.worst + smoothness.worst +
    compactness.worst + concavity.worst + concavepoints.worst +
    symmetry.worst + fractaldimension.worst

                         Df Deviance    AIC
- texture                 1   118.69 176.69
- smoothness.stderr       1   118.80 176.80
- perimeter.stderr        1   118.81 176.81
- concavepoints.stderr    1   118.85 176.85
- area.stderr             1   118.86 176.86
- perimeter               1   118.88 176.88
- concavepoints.worst     1   118.94 176.94
- symmetry.worst          1   119.02 177.01
- smoothness              1   119.12 177.12
- fractaldimension.stderr 1   119.18 177.18
- symmetry.stderr         1   119.21 177.21
- fractaldimension        1   119.38 177.38
- perimeter.worst         1   119.41 177.41
- fractaldimension.worst  1   119.62 177.62
- compactness.stderr      1   119.73 177.73
- id                      1   119.82 177.82
- compactness.worst       1   120.28 178.28
- smoothness.worst        1   120.29 178.29
<none>                        118.67 178.67
- texture.stderr          1   120.69 178.69
- radius.stderr           1   121.38 179.38
- radius                  1   121.80 179.80
- radius.worst            1   122.05 180.05
- concavity.worst         1   123.12 181.12
- texture.worst           1   123.96 181.96
- concavity               1   124.22 182.22
- area                    1   124.72 182.72
- concavity.stderr        1   126.06 184.06
- concavepoints           1   127.02 185.02
- area.worst              1   132.05 190.05

Step:  AIC=176.7
```

```
diagnosis ~ id + radius + perimeter + area + smoothness + concavity +
    concavepoints + fractaldimension + radius.stderr + texture.stderr +
    perimeter.stderr + area.stderr + smoothness.stderr + compactness.stderr +
    concavity.stderr + concavepoints.stderr + symmetry.stderr +
    fractaldimension.stderr + radius.worst + texture.worst +
    perimeter.worst + area.worst + smoothness.worst + compactness.worst +
    concavity.worst + concavepoints.worst + symmetry.worst +
    fractaldimension.worst

                         Df Deviance    AIC
- smoothness.stderr       1   118.82 174.82
- perimeter.stderr        1   118.85 174.85
- concavepoints.stderr    1   118.89 174.89
- area.stderr             1   118.89 174.89
- perimeter               1   118.90 174.90
- concavepoints.worst     1   118.95 174.95
- smoothness              1   119.15 175.15
- symmetry.worst          1   119.16 175.16
- fractaldimension.stderr 1   119.20 175.20
- symmetry.stderr         1   119.21 175.21
- fractaldimension        1   119.40 175.40
- perimeter.worst         1   119.42 175.42
- fractaldimension.worst  1   119.62 175.62
- compactness.stderr      1   119.73 175.73
- id                      1   119.88 175.88
- smoothness.worst        1   120.41 176.41
- compactness.worst       1   120.42 176.42
<none>                        118.69 176.69
- texture.stderr          1   121.00 177.00
- radius.stderr           1   121.40 177.40
- radius                  1   121.85 177.85
- radius.worst            1   122.22 178.22
- concavity.worst         1   123.28 179.28
- concavity               1   124.27 180.27
- area                    1   125.19 181.19
- concavity.stderr        1   126.21 182.21
- concavepoints           1   127.04 183.04
- area.worst              1   132.26 188.26
- texture.worst           1   142.25 198.25

Step:  AIC=174.82
diagnosis ~ id + radius + perimeter + area + smoothness + concavity +
    concavepoints + fractaldimension + radius.stderr + texture.stderr +
```

```
    perimeter.stderr + area.stderr + compactness.stderr + concavity.stderr +
    concavepoints.stderr + symmetry.stderr + fractaldimension.stderr +
    radius.worst + texture.worst + perimeter.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- concavepoints.stderr     1   118.89 172.89
- area.stderr              1   119.00 173.00
- perimeter                1   119.00 173.00
- perimeter.stderr         1   119.07 173.07
- symmetry.worst           1   119.26 173.26
- symmetry.stderr          1   119.34 173.34
- concavepoints.worst      1   119.35 173.35
- smoothness               1   119.40 173.40
- fractaldimension.stderr  1   119.41 173.41
- fractaldimension         1   119.59 173.59
- perimeter.worst          1   119.68 173.68
- compactness.stderr       1   119.74 173.74
- fractaldimension.worst   1   119.86 173.86
- id                       1   120.58 174.58
- compactness.worst        1   120.75 174.75
<none>                         118.82 174.82
- texture.stderr           1   121.08 175.09
- radius.stderr            1   121.77 175.77
- radius                   1   121.88 175.88
- radius.worst             1   122.23 176.23
- smoothness.worst         1   122.69 176.69
- concavity.worst          1   123.47 177.47
- concavity                1   124.31 178.31
- area                     1   125.31 179.31
- concavity.stderr         1   126.35 180.35
- concavepoints            1   127.24 181.24
- area.worst               1   132.31 186.31
- texture.worst            1   142.69 196.69

Step:  AIC=172.89
diagnosis ~ id + radius + perimeter + area + smoothness + concavity +
    concavepoints + fractaldimension + radius.stderr + texture.stderr +
    perimeter.stderr + area.stderr + compactness.stderr + concavity.stderr +
    symmetry.stderr + fractaldimension.stderr + radius.worst +
    texture.worst + perimeter.worst + area.worst + smoothness.worst +
    compactness.worst + concavity.worst + concavepoints.worst +
```

```
    symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- area.stderr              1   119.05 171.05
- perimeter                1   119.05 171.05
- perimeter.stderr         1   119.23 171.24
- symmetry.stderr          1   119.39 171.39
- symmetry.worst           1   119.39 171.39
- smoothness               1   119.50 171.50
- fractaldimension.stderr  1   119.53 171.53
- fractaldimension         1   119.64 171.64
- perimeter.worst          1   119.84 171.84
- fractaldimension.worst   1   120.03 172.03
- compactness.stderr       1   120.15 172.15
- concavepoints.worst      1   120.27 172.27
- id                       1   120.59 172.59
- compactness.worst        1   120.77 172.76
<none>                         118.89 172.89
- texture.stderr           1   121.09 173.09
- radius                   1   122.02 174.01
- radius.stderr            1   122.09 174.09
- radius.worst             1   122.26 174.26
- smoothness.worst         1   122.76 174.76
- concavity.worst          1   124.09 176.09
- concavity                1   124.74 176.74
- area                     1   125.69 177.69
- concavity.stderr         1   127.67 179.67
- concavepoints            1   127.83 179.83
- area.worst               1   132.38 184.38
- texture.worst            1   142.69 194.69

Step:  AIC=171.05
diagnosis ~ id + radius + perimeter + area + smoothness + concavity +
    concavepoints + fractaldimension + radius.stderr + texture.stderr +
    perimeter.stderr + compactness.stderr + concavity.stderr +
    symmetry.stderr + fractaldimension.stderr + radius.worst +
    texture.worst + perimeter.worst + area.worst + smoothness.worst +
    compactness.worst + concavity.worst + concavepoints.worst +
    symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- perimeter                1   119.21 169.21
- perimeter.stderr         1   119.43 169.43
```

```
- symmetry.worst           1   119.48 169.48
- symmetry.stderr          1   119.60 169.60
- smoothness               1   119.64 169.64
- fractaldimension.stderr  1   119.65 169.65
- fractaldimension         1   119.79 169.79
- perimeter.worst          1   120.02 170.02
- fractaldimension.worst   1   120.11 170.12
- compactness.stderr       1   120.23 170.23
- concavepoints.worst      1   120.45 170.45
- id                       1   120.66 170.66
- compactness.worst        1   120.97 170.97
<none>                         119.05 171.05
- texture.stderr           1   121.12 171.12
- radius                   1   122.17 172.17
- radius.worst             1   122.26 172.26
- smoothness.worst         1   122.96 172.96
- radius.stderr            1   123.03 173.03
- concavity.worst          1   124.68 174.68
- concavity                1   125.24 175.24
- area                     1   127.61 177.61
- concavity.stderr         1   127.70 177.71
- concavepoints            1   128.15 178.15
- area.worst               1   140.43 190.43
- texture.worst            1   142.71 192.71

Step:  AIC=169.21
diagnosis ~ id + radius + area + smoothness + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + perimeter.stderr +
    compactness.stderr + concavity.stderr + symmetry.stderr +
    fractaldimension.stderr + radius.worst + texture.worst +
    perimeter.worst + area.worst + smoothness.worst + compactness.worst +
    concavity.worst + concavepoints.worst + symmetry.worst +
    fractaldimension.worst

                          Df Deviance    AIC
- perimeter.stderr         1   119.62 167.62
- symmetry.stderr          1   119.69 167.69
- symmetry.worst           1   119.69 167.69
- smoothness               1   119.77 167.77
- fractaldimension.stderr  1   119.90 167.90
- fractaldimension         1   120.00 168.00
- perimeter.worst          1   120.22 168.22
- compactness.stderr       1   120.23 168.23
```

```
- fractaldimension.worst   1   120.30 168.30
- concavepoints.worst      1   120.61 168.61
- id                       1   120.79 168.79
- compactness.worst        1   121.16 169.16
<none>                         119.21 169.21
- texture.stderr           1   121.33 169.33
- radius.worst             1   122.85 170.85
- smoothness.worst         1   122.98 170.98
- radius.stderr            1   123.34 171.34
- radius                   1   124.38 172.38
- concavity.worst          1   124.80 172.80
- concavity                1   125.97 173.97
- area                     1   127.82 175.82
- concavity.stderr         1   127.94 175.94
- concavepoints            1   128.24 176.24
- area.worst               1   142.31 190.31
- texture.worst            1   142.72 190.72

Step:  AIC=167.62
diagnosis ~ id + radius + area + smoothness + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + symmetry.stderr + fractaldimension.stderr +
    radius.worst + texture.worst + perimeter.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- symmetry.stderr          1   119.99 165.99
- smoothness               1   120.10 166.10
- fractaldimension.stderr  1   120.10 166.10
- symmetry.worst           1   120.19 166.19
- perimeter.worst          1   120.33 166.33
- fractaldimension.worst   1   120.63 166.63
- fractaldimension         1   120.74 166.74
- compactness.stderr       1   120.89 166.90
- concavepoints.worst      1   121.04 167.04
- id                       1   121.10 167.10
- compactness.worst        1   121.20 167.20
<none>                         119.62 167.62
- texture.stderr           1   122.23 168.23
- smoothness.worst         1   123.25 169.25
- radius                   1   124.39 170.39
- concavity.worst          1   124.83 170.83
```

```
- concavity                1   125.99 171.99
- area                     1   127.87 173.87
- concavity.stderr         1   128.29 174.29
- concavepoints            1   128.33 174.33
- radius.worst             1   129.16 175.16
- area.worst               1   142.43 188.43
- texture.worst            1   143.87 189.87
- radius.stderr            1   153.30 199.30

Step:  AIC=165.99
diagnosis ~ id + radius + area + smoothness + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + fractaldimension.stderr + radius.worst +
    texture.worst + perimeter.worst + area.worst + smoothness.worst +
    compactness.worst + concavity.worst + concavepoints.worst +
    symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- smoothness               1   120.39 164.39
- fractaldimension.stderr  1   120.43 164.43
- perimeter.worst          1   120.59 164.59
- fractaldimension.worst   1   120.83 164.83
- compactness.stderr       1   120.90 164.90
- fractaldimension         1   120.97 164.97
- id                       1   121.24 165.24
- concavepoints.worst      1   121.25 165.25
<none>                         119.99 165.99
- compactness.worst        1   122.10 166.10
- texture.stderr           1   122.58 166.58
- smoothness.worst         1   123.33 167.33
- radius                   1   124.71 168.71
- symmetry.worst           1   124.72 168.72
- concavity.worst          1   124.93 168.93
- concavity                1   125.99 169.99
- area                     1   127.88 171.88
- concavity.stderr         1   128.35 172.35
- concavepoints            1   128.59 172.59
- radius.worst             1   129.17 173.17
- area.worst               1   142.48 186.48
- texture.worst            1   143.95 187.95
- radius.stderr            1   155.26 199.26

Step:  AIC=164.39
```

```
diagnosis ~ id + radius + area + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + fractaldimension.stderr + radius.worst +
    texture.worst + perimeter.worst + area.worst + smoothness.worst +
    compactness.worst + concavity.worst + concavepoints.worst +
    symmetry.worst + fractaldimension.worst

                          Df Deviance    AIC
- fractaldimension.stderr  1   120.74 162.74
- perimeter.worst          1   121.00 163.00
- id                       1   121.37 163.37
- fractaldimension.worst   1   121.37 163.37
- compactness.stderr       1   121.37 163.37
- concavepoints.worst      1   121.43 163.43
- fractaldimension         1   122.07 164.07
<none>                         120.39 164.39
- compactness.worst        1   122.54 164.54
- texture.stderr           1   123.11 165.12
- smoothness.worst         1   124.10 166.10
- symmetry.worst           1   124.85 166.85
- radius                   1   124.93 166.93
- concavity.worst          1   125.14 167.14
- concavity                1   126.01 168.01
- area                     1   127.89 169.89
- concavity.stderr         1   128.47 170.47
- concavepoints            1   129.13 171.13
- radius.worst             1   129.45 171.45
- area.worst               1   142.50 184.50
- texture.worst            1   146.60 188.60
- radius.stderr            1   156.19 198.19

Step:  AIC=162.74
diagnosis ~ id + radius + area + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + radius.worst + texture.worst + perimeter.worst +
    area.worst + smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                         Df Deviance    AIC
- perimeter.worst         1   121.23 161.23
- fractaldimension.worst  1   121.37 161.37
- id                      1   121.67 161.67
- concavepoints.worst     1   122.01 162.01
```

```
- fractaldimension        1   122.42 162.43
- compactness.worst       1   122.55 162.55
<none>                        120.74 162.74
- texture.stderr          1   123.81 163.81
- compactness.stderr      1   124.20 164.20
- smoothness.worst        1   124.39 164.39
- symmetry.worst          1   125.41 165.41
- radius                  1   125.42 165.42
- concavity               1   127.11 167.11
- concavity.worst         1   128.16 168.16
- area                    1   128.81 168.81
- concavity.stderr        1   129.44 169.44
- concavepoints           1   129.82 169.82
- radius.worst            1   130.14 170.14
- area.worst              1   142.55 182.55
- texture.worst           1   147.31 187.31
- radius.stderr           1   157.69 197.69

Step:  AIC=161.23
diagnosis ~ id + radius + area + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + radius.worst + texture.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst + fractaldimension.worst

                         Df Deviance    AIC
- fractaldimension.worst  1   121.86 159.86
- id                      1   122.26 160.26
- concavepoints.worst     1   122.34 160.34
- compactness.worst       1   122.61 160.61
<none>                        121.23 161.23
- fractaldimension        1   123.31 161.31
- texture.stderr          1   124.35 162.35
- smoothness.worst        1   124.45 162.46
- compactness.stderr      1   124.73 162.73
- symmetry.worst          1   125.46 163.46
- radius                  1   125.88 163.88
- concavity               1   127.11 165.11
- concavity.worst         1   128.32 166.32
- area                    1   129.12 167.12
- concavity.stderr        1   129.51 167.51
- concavepoints           1   130.24 168.24
- radius.worst            1   138.91 176.91
```

```
- area.worst              1   142.59 180.59
- texture.worst           1   147.46 185.46
- radius.stderr           1   157.70 195.70

Step:  AIC=159.86
diagnosis ~ id + radius + area + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + radius.worst + texture.worst + area.worst +
    smoothness.worst + compactness.worst + concavity.worst +
    concavepoints.worst + symmetry.worst

                      Df Deviance    AIC
- compactness.worst    1   122.69 158.69
- concavepoints.worst  1   122.73 158.73
- id                   1   122.76 158.76
- fractaldimension     1   123.37 159.37
<none>                     121.86 159.86
- texture.stderr       1   124.93 160.93
- smoothness.worst     1   125.28 161.28
- compactness.stderr   1   125.36 161.36
- symmetry.worst       1   125.80 161.80
- radius               1   126.98 162.99
- concavity            1   127.94 163.94
- concavity.worst      1   129.36 165.36
- area                 1   129.66 165.66
- concavepoints        1   130.26 166.26
- concavity.stderr     1   130.49 166.49
- radius.worst         1   139.57 175.57
- area.worst           1   142.83 178.83
- texture.worst        1   148.78 184.78
- radius.stderr        1   158.06 194.06

Step:  AIC=158.7
diagnosis ~ id + radius + area + concavity + concavepoints +
    fractaldimension + radius.stderr + texture.stderr + compactness.stderr +
    concavity.stderr + radius.worst + texture.worst + area.worst +
    smoothness.worst + concavity.worst + concavepoints.worst +
    symmetry.worst

                      Df Deviance    AIC
- id                   1   123.69 157.69
- concavepoints.worst  1   124.14 158.13
<none>                     122.69 158.69
```

```
- fractaldimension     1   125.58 159.58
- texture.stderr       1   125.61 159.61
- smoothness.worst     1   126.07 160.07
- symmetry.worst       1   126.12 160.12
- concavity            1   128.87 162.87
- radius               1   129.01 163.01
- concavity.worst      1   129.57 163.57
- area                 1   129.92 163.92
- compactness.stderr   1   130.26 164.26
- concavepoints        1   131.94 165.94
- concavity.stderr     1   136.22 170.22
- radius.worst         1   139.63 173.63
- area.worst           1   143.02 177.02
- texture.worst        1   149.27 183.27
- radius.stderr        1   158.78 192.78

Step:  AIC=157.69
diagnosis ~ radius + area + concavity + concavepoints + fractaldimension +
    radius.stderr + texture.stderr + compactness.stderr + concavity.stderr +
    radius.worst + texture.worst + area.worst + smoothness.worst +
    concavity.worst + concavepoints.worst + symmetry.worst

                      Df Deviance    AIC
- concavepoints.worst  1   124.88 156.88
<none>                     123.69 157.69
- symmetry.worst       1   126.62 158.62
- fractaldimension     1   126.87 158.87
- texture.stderr       1   127.24 159.24
- smoothness.worst     1   128.24 160.24
- concavity            1   129.07 161.07
- concavity.worst      1   129.57 161.57
- radius               1   130.09 162.09
- area                 1   130.24 162.24
- compactness.stderr   1   130.27 162.27
- concavepoints        1   132.15 164.15
- concavity.stderr     1   137.37 169.37
- radius.worst         1   140.29 172.29
- area.worst           1   144.04 176.04
- texture.worst        1   151.89 183.89
- radius.stderr        1   159.76 191.76

Step:  AIC=156.88
diagnosis ~ radius + area + concavity + concavepoints + fractaldimension +
```

```
    radius.stderr + texture.stderr + compactness.stderr + concavity.stderr +
    radius.worst + texture.worst + area.worst + smoothness.worst +
    concavity.worst + symmetry.worst

                     Df Deviance    AIC
<none>                    124.88 156.88
- symmetry.worst      1   127.63 157.63
- fractaldimension    1   127.76 157.76
- texture.stderr      1   127.87 157.87
- smoothness.worst    1   128.77 158.77
- concavity           1   129.12 159.12
- concavity.worst     1   129.68 159.68
- radius              1   130.24 160.24
- area                1   130.30 160.30
- compactness.stderr  1   131.53 161.53
- concavepoints       1   132.94 162.94
- concavity.stderr    1   137.39 167.39
- radius.worst        1   140.96 170.96
- area.worst          1   145.80 175.80
- texture.worst       1   152.01 182.01
- radius.stderr       1   161.09 191.09
```

```r
# Fit null model
null.model <- glm(diagnosis ~ 1, data = wdbc2, family = "binomial")



# Perform forward stepwise selection
sel.forw <-
  stepAIC(null.model,
          scope = list(upper = full.model),
          direction = "forward")
```

```
Start:  AIC=751.33
diagnosis ~ 1

                        Df Deviance    AIC
+ perimeter.worst         1   305.54 309.54
+ concavepoints.worst     1   311.45 315.45
+ radius.worst            1   317.15 321.15
+ concavepoints           1   320.97 324.97
```

```
+ area.worst               1   339.18 343.18
+ perimeter                1   368.15 372.15
+ radius                   1   380.32 384.32
+ area                     1   406.07 410.07
+ area.stderr              1   410.88 414.88
+ concavity                1   411.06 415.06
+ concavity.worst          1   459.05 463.05
+ radius.stderr            1   487.88 491.88
+ perimeter.stderr         1   489.30 493.30
+ compactness              1   528.91 532.91
+ compactness.worst        1   531.63 535.63
+ texture.worst            1   631.70 635.70
+ concavepoints.stderr     1   639.50 643.50
+ smoothness.worst         1   646.28 650.28
+ symmetry.worst           1   647.53 651.53
+ texture                  1   651.22 655.22
+ smoothness               1   673.44 677.44
+ symmetry                 1   687.56 691.56
+ fractaldimension.worst   1   690.34 694.34
+ concavity.stderr         1   692.64 696.64
+ compactness.stderr       1   699.78 703.78
+ fractaldimension.stderr  1   742.26 746.26
+ smoothness.stderr        1   746.97 750.97
<none>                        749.33 751.33
+ id                       1   748.49 752.49
+ texture.stderr           1   749.25 753.25
+ symmetry.stderr          1   749.32 753.32
+ fractaldimension         1   749.32 753.32

Step:  AIC=309.54
diagnosis ~ perimeter.worst

                          Df Deviance    AIC
+ smoothness.worst         1   250.78 256.78
+ concavepoints.worst      1   252.87 258.87
+ concavity.worst          1   264.93 270.93
+ concavepoints            1   265.99 271.99
+ concavity                1   266.82 272.82
+ smoothness               1   268.46 274.46
+ texture.worst            1   269.36 275.36
+ fractaldimension.worst   1   270.53 276.53
+ symmetry.worst           1   271.04 277.04
+ area                     1   271.15 277.15
```

```
+ fractaldimension         1   276.19 282.19
+ compactness.worst        1   276.32 282.32
+ symmetry                 1   276.71 282.71
+ compactness              1   277.36 283.36
+ texture                  1   281.37 287.37
+ concavepoints.stderr     1   288.81 294.81
+ concavity.stderr         1   290.13 296.13
+ perimeter                1   291.26 297.26
+ radius.stderr            1   292.41 298.41
+ texture.stderr           1   292.43 298.43
+ area.worst               1   293.50 299.50
+ smoothness.stderr        1   294.18 300.18
+ fractaldimension.stderr  1   295.74 301.74
+ compactness.stderr       1   296.85 302.85
+ perimeter.stderr         1   297.25 303.25
+ radius                   1   297.29 303.29
+ symmetry.stderr          1   298.41 304.41
+ area.stderr              1   300.85 306.85
<none>                        305.54 309.54
+ radius.worst             1   305.15 311.15
+ id                       1   305.54 311.54

Step:  AIC=256.78
diagnosis ~ perimeter.worst + smoothness.worst

                          Df Deviance    AIC
+ texture                  1   227.48 235.48
+ texture.worst            1   228.31 236.31
+ radius.stderr            1   235.68 243.68
+ texture.stderr           1   236.80 244.80
+ concavity.stderr         1   239.02 247.02
+ concavity                1   241.42 249.42
+ perimeter.stderr         1   241.91 249.91
+ concavepoints.stderr     1   242.42 250.42
+ concavity.worst          1   243.07 251.07
+ area                     1   243.09 251.09
+ area.stderr              1   243.16 251.16
+ concavepoints.worst      1   243.98 251.98
+ symmetry.worst           1   245.05 253.05
+ symmetry                 1   246.31 254.31
+ concavepoints            1   246.45 254.45
+ symmetry.stderr          1   246.59 254.59
+ radius                   1   247.13 255.13
```

```
+ area.worst               1   247.15 255.15
+ fractaldimension.stderr  1   247.47 255.47
<none>                         250.78 256.78
+ compactness.stderr       1   249.00 257.00
+ radius.worst             1   249.31 257.31
+ fractaldimension.worst   1   249.45 257.45
+ compactness.worst        1   249.82 257.82
+ compactness              1   250.07 258.07
+ fractaldimension         1   250.56 258.56
+ smoothness.stderr        1   250.62 258.62
+ perimeter                1   250.73 258.73
+ id                       1   250.75 258.75
+ smoothness               1   250.77 258.77

Step:  AIC=235.48
diagnosis ~ perimeter.worst + smoothness.worst + texture

                         Df Deviance    AIC
+ radius.stderr           1   215.29 225.29
+ concavity.stderr        1   215.96 225.96
+ concavity               1   219.21 229.21
+ concavepoints.stderr    1   219.70 229.70
+ concavepoints.worst     1   219.95 229.95
+ area                    1   220.66 230.66
+ perimeter.stderr        1   220.92 230.92
+ concavepoints           1   220.97 230.97
+ radius                  1   221.13 231.13
+ symmetry                1   221.17 231.17
+ concavity.worst         1   221.18 231.18
+ symmetry.worst          1   221.56 231.56
+ area.stderr             1   222.17 232.17
+ area.worst              1   222.76 232.76
+ symmetry.stderr         1   224.14 234.14
+ texture.stderr          1   224.78 234.78
+ radius.worst            1   225.21 235.21
+ fractaldimension.stderr 1   225.26 235.26
<none>                       227.48 235.48
+ smoothness              1   225.72 235.72
+ compactness             1   226.62 236.62
+ texture.worst           1   226.75 236.75
+ fractaldimension.worst  1   226.83 236.83
+ fractaldimension        1   226.99 236.99
+ compactness.stderr      1   227.03 237.03
```

```
+ compactness.worst        1   227.23 237.23
+ smoothness.stderr        1   227.29 237.29
+ id                       1   227.42 237.42
+ perimeter                1   227.43 237.43

Step:  AIC=225.29
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr

                          Df Deviance    AIC
+ area.stderr              1   199.84 211.84
+ concavepoints.worst      1   204.57 216.57
+ area.worst               1   205.62 217.62
+ concavity.worst          1   206.07 218.07
+ symmetry.worst           1   206.20 218.20
+ area                     1   206.49 218.49
+ radius                   1   208.52 220.52
+ concavity.stderr         1   209.01 221.01
+ concavity                1   210.26 222.26
+ smoothness.stderr        1   211.62 223.62
+ texture.worst            1   211.95 223.95
+ symmetry                 1   212.46 224.46
+ perimeter.stderr         1   212.55 224.55
+ concavepoints            1   212.90 224.90
<none>                        215.29 225.29
+ fractaldimension.worst   1   213.60 225.60
+ concavepoints.stderr     1   213.63 225.63
+ compactness.worst        1   213.86 225.86
+ radius.worst             1   214.43 226.43
+ symmetry.stderr          1   214.94 226.94
+ compactness              1   215.07 227.07
+ fractaldimension.stderr  1   215.14 227.14
+ id                       1   215.15 227.15
+ smoothness               1   215.25 227.25
+ perimeter                1   215.25 227.25
+ texture.stderr           1   215.28 227.28
+ fractaldimension         1   215.28 227.28
+ compactness.stderr       1   215.29 227.29

Step:  AIC=211.84
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr

                          Df Deviance    AIC
```

```
+ radius                    1   192.42 206.42
+ symmetry.worst            1   193.32 207.32
+ concavity.worst           1   193.38 207.38
+ concavepoints.worst       1   193.39 207.39
+ area                      1   194.93 208.93
+ texture.worst             1   195.54 209.54
+ concavity.stderr          1   195.54 209.54
+ perimeter.stderr          1   196.09 210.09
+ concavity                 1   196.70 210.70
+ radius.worst              1   197.16 211.16
+ smoothness.stderr         1   197.58 211.58
<none>                          199.84 211.84
+ symmetry                  1   198.17 212.17
+ area.worst                1   198.50 212.50
+ concavepoints             1   198.67 212.67
+ concavepoints.stderr      1   199.15 213.15
+ fractaldimension.worst    1   199.33 213.33
+ smoothness                1   199.52 213.52
+ symmetry.stderr           1   199.54 213.54
+ perimeter                 1   199.54 213.54
+ compactness.worst         1   199.56 213.56
+ fractaldimension          1   199.59 213.59
+ compactness               1   199.81 213.81
+ compactness.stderr        1   199.82 213.82
+ id                        1   199.83 213.83
+ fractaldimension.stderr   1   199.83 213.83
+ texture.stderr            1   199.84 213.84

Step:  AIC=206.42
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius

                          Df Deviance    AIC
+ symmetry.worst           1   181.05 197.05
+ concavity.worst          1   183.30 199.30
+ texture.worst            1   185.15 201.15
+ area                     1   185.42 201.42
+ concavepoints.worst      1   185.57 201.57
+ smoothness.stderr        1   188.03 204.03
+ concavity.stderr         1   189.59 205.59
+ area.worst               1   189.79 205.79
+ perimeter.stderr         1   190.15 206.15
+ concavity                1   190.18 206.18
```

```
<none>                          192.42 206.42
+ symmetry                 1    190.52 206.52
+ fractaldimension.worst   1    190.68 206.68
+ compactness.worst        1    191.05 207.05
+ perimeter                1    191.61 207.61
+ radius.worst             1    191.72 207.72
+ concavepoints            1    191.84 207.84
+ smoothness               1    191.95 207.95
+ symmetry.stderr          1    192.16 208.16
+ id                       1    192.29 208.29
+ texture.stderr           1    192.34 208.34
+ fractaldimension.stderr  1    192.38 208.38
+ compactness              1    192.39 208.39
+ compactness.stderr       1    192.40 208.40
+ fractaldimension         1    192.40 208.40
+ concavepoints.stderr     1    192.42 208.42

Step:  AIC=197.05
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst

                         Df Deviance    AIC
+ texture.worst           1    175.04 193.04
+ symmetry.stderr         1    175.72 193.72
+ concavity.worst         1    176.47 194.47
+ area                    1    177.03 195.03
+ concavepoints.worst     1    177.59 195.59
+ area.worst              1    178.69 196.69
+ smoothness.stderr       1    179.04 197.04
<none>                          181.05 197.05
+ perimeter.stderr        1    179.10 197.10
+ compactness             1    179.72 197.72
+ symmetry                1    179.87 197.87
+ radius.worst            1    179.89 197.89
+ fractaldimension        1    179.91 197.91
+ concavity.stderr        1    179.93 197.93
+ compactness.stderr      1    180.05 198.05
+ fractaldimension.stderr 1    180.43 198.43
+ concavity               1    180.48 198.48
+ smoothness              1    180.56 198.56
+ concavepoints           1    180.75 198.75
+ perimeter               1    180.94 198.94
+ compactness.worst       1    180.98 198.98
```

```
+ texture.stderr           1   180.99 198.99
+ id                       1   180.99 198.99
+ concavepoints.stderr     1   181.01 199.01
+ fractaldimension.worst   1   181.03 199.03

Step:  AIC=193.04
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst

                         Df Deviance    AIC
+ area.worst              1   168.87 188.87
+ concavity.worst         1   170.07 190.07
+ concavepoints.worst     1   171.18 191.18
+ area                    1   171.77 191.77
+ symmetry.stderr         1   172.59 192.59
+ texture.stderr          1   172.92 192.92
<none>                         175.04 193.04
+ concavity.stderr        1   173.72 193.72
+ concavepoints           1   173.98 193.98
+ symmetry                1   173.99 193.99
+ concavity               1   174.23 194.23
+ perimeter.stderr        1   174.25 194.25
+ fractaldimension        1   174.41 194.41
+ smoothness.stderr       1   174.52 194.52
+ compactness             1   174.59 194.59
+ fractaldimension.stderr 1   174.63 194.63
+ compactness.stderr      1   174.79 194.79
+ perimeter               1   174.83 194.83
+ radius.worst            1   174.96 194.96
+ fractaldimension.worst  1   175.01 195.01
+ smoothness              1   175.01 195.01
+ compactness.worst       1   175.01 195.01
+ concavepoints.stderr    1   175.01 195.01
+ id                      1   175.03 195.03

Step:  AIC=188.87
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst

                         Df Deviance    AIC
+ radius.worst            1   159.32 181.32
+ perimeter.stderr        1   161.90 183.90
+ symmetry.stderr         1   163.91 185.91
```

```
+ texture.stderr           1   164.07 186.07
+ compactness.stderr       1   164.86 186.86
+ compactness              1   165.87 187.87
+ concavity.worst          1   166.51 188.51
<none>                         168.87 188.87
+ fractaldimension         1   166.95 188.95
+ smoothness.stderr        1   167.18 189.18
+ compactness.worst        1   167.20 189.20
+ concavepoints.worst      1   167.36 189.36
+ area                     1   167.44 189.44
+ fractaldimension.stderr  1   167.72 189.72
+ symmetry                 1   167.72 189.72
+ concavity.stderr         1   168.23 190.23
+ smoothness               1   168.38 190.38
+ perimeter                1   168.64 190.64
+ concavepoints.stderr     1   168.66 190.66
+ fractaldimension.worst   1   168.71 190.71
+ concavity                1   168.81 190.81
+ concavepoints            1   168.84 190.84
+ id                       1   168.87 190.87

Step:  AIC=181.32
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst +
    radius.worst

                         Df Deviance    AIC
+ concavity.worst          1   149.76 173.76
+ concavity.stderr         1   153.12 177.12
+ concavepoints.worst      1   154.55 178.55
+ concavity                1   155.32 179.32
+ symmetry.stderr          1   156.72 180.72
+ area                     1   156.97 180.97
<none>                         159.32 181.32
+ perimeter                1   157.77 181.77
+ perimeter.stderr         1   157.93 181.93
+ concavepoints.stderr     1   158.19 182.19
+ texture.stderr           1   158.20 182.20
+ compactness.stderr       1   158.21 182.21
+ smoothness.stderr        1   158.46 182.46
+ concavepoints            1   158.70 182.70
+ smoothness               1   158.86 182.86
+ compactness              1   158.90 182.90
```

```
+ fractaldimension.worst   1   159.15 183.15
+ fractaldimension         1   159.16 183.16
+ compactness.worst        1   159.18 183.18
+ symmetry                 1   159.20 183.20
+ id                       1   159.32 183.32
+ fractaldimension.stderr  1   159.32 183.32

Step:  AIC=173.76
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst +
    radius.worst + concavity.worst

                          Df Deviance    AIC
+ compactness.stderr       1   139.36 165.36
+ compactness.worst        1   142.83 168.83
+ compactness              1   144.48 170.48
+ fractaldimension         1   146.28 172.28
+ area                     1   146.35 172.35
+ fractaldimension.stderr  1   146.76 172.76
+ symmetry.stderr          1   147.29 173.29
+ texture.stderr           1   147.46 173.46
<none>                         149.76 173.76
+ fractaldimension.worst   1   147.88 173.88
+ perimeter                1   148.30 174.30
+ smoothness.stderr        1   148.86 174.86
+ concavity                1   149.10 175.10
+ symmetry                 1   149.24 175.24
+ concavepoints.stderr     1   149.38 175.38
+ smoothness               1   149.47 175.47
+ concavepoints.worst      1   149.52 175.52
+ concavity.stderr         1   149.56 175.56
+ concavepoints            1   149.57 175.57
+ perimeter.stderr         1   149.75 175.75
+ id                       1   149.75 175.75

Step:  AIC=165.36
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst +
    radius.worst + concavity.worst + compactness.stderr

                          Df Deviance    AIC
+ concavity.stderr         1   136.72 164.72
+ compactness.worst        1   136.95 164.95
```

```
<none>                          139.36 165.36
+ concavepoints.stderr     1   137.78 165.78
+ id                       1   138.49 166.49
+ area                     1   138.59 166.59
+ compactness              1   138.62 166.62
+ smoothness.stderr        1   138.77 166.77
+ texture.stderr           1   138.82 166.82
+ symmetry                 1   138.88 166.88
+ fractaldimension.stderr  1   138.98 166.98
+ perimeter.stderr         1   139.00 167.00
+ perimeter                1   139.01 167.01
+ concavepoints.worst      1   139.03 167.03
+ fractaldimension         1   139.03 167.03
+ fractaldimension.worst   1   139.10 167.10
+ concavepoints            1   139.15 167.15
+ concavity                1   139.31 167.31
+ symmetry.stderr          1   139.36 167.36
+ smoothness               1   139.36 167.36

Step:  AIC=164.71
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst +
    radius.worst + concavity.worst + compactness.stderr + concavity.stderr

                         Df Deviance    AIC
+ texture.stderr          1   132.69 162.69
<none>                       136.72 164.72
+ compactness             1   135.12 165.12
+ area                    1   135.31 165.31
+ fractaldimension        1   135.78 165.78
+ symmetry                1   135.94 165.94
+ id                      1   136.02 166.02
+ compactness.worst       1   136.02 166.02
+ fractaldimension.stderr 1   136.09 166.09
+ concavity               1   136.11 166.11
+ concavepoints.worst     1   136.42 166.42
+ perimeter               1   136.51 166.51
+ fractaldimension.worst  1   136.59 166.59
+ smoothness.stderr       1   136.59 166.59
+ concavepoints.stderr    1   136.65 166.65
+ concavepoints           1   136.68 166.68
+ symmetry.stderr         1   136.70 166.70
+ perimeter.stderr        1   136.70 166.70
```

```
+ smoothness               1   136.71 166.71

Step:  AIC=162.69
diagnosis ~ perimeter.worst + smoothness.worst + texture + radius.stderr +
    area.stderr + radius + symmetry.worst + texture.worst + area.worst +
    radius.worst + concavity.worst + compactness.stderr + concavity.stderr +
    texture.stderr

                         Df Deviance    AIC
<none>                        132.69 162.69
+ concavepoints           1   131.03 163.03
+ compactness.worst       1   131.28 163.28
+ symmetry.stderr         1   131.82 163.82
+ compactness             1   132.21 164.21
+ concavepoints.worst     1   132.21 164.21
+ smoothness              1   132.34 164.34
+ id                      1   132.40 164.40
+ perimeter.stderr        1   132.40 164.40
+ fractaldimension.worst  1   132.41 164.41
+ fractaldimension.stderr 1   132.44 164.44
+ fractaldimension        1   132.44 164.44
+ concavepoints.stderr    1   132.44 164.44
+ smoothness.stderr       1   132.53 164.53
+ area                    1   132.61 164.61
+ perimeter               1   132.64 164.64
+ symmetry                1   132.66 164.66
+ concavity               1   132.69 164.69
```

```r
# Load the dplyr library
library(dplyr)
```

```
Attaching package: 'dplyr'

The following object is masked from 'package:kableExtra':

    group_rows

The following object is masked from 'package:MASS':

    select
```

```
The following objects are masked from 'package:data.table':

    between, first, last

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
sel.back.names <- names(sel.back$model)[-1] # exclude intercept
x.sel.back <- x.wdbc2[, sel.back.names]



# Fit the forward regression model with standardized predictor variables
sel.forw.coef <- coef(sel.forw, standardize = TRUE)
sel.forw.names <- names(sel.forw$model)[-1] # exclude intercept
x.sel.forw <- x.wdbc2[, sel.forw.names]


# Fit the forward regression model with standardized predictor variables
sel.forw.coef <- coef(sel.forw, standardize = TRUE)
sel.back.coef <- coef(sel.back, standardize = TRUE)

# Create dataframe of results for backward and forward selection
results <- data.frame(
  Variable = c(sel.back.names, sel.forw.names),
  Method = c(rep("B", length(sel.back.names)), rep("S", length(sel.forw.names))),
  Coefficient = c(sel.back.coef[-1], sel.forw.coef[-1])
) %>%
  arrange(abs(Coefficient)) # sort by increasing absolute value of standardized coefficien

# Print results as table without including standard deviation
kable(
  results,
  digits = 4,
  align = "c",
  col.names = c("Variable", "Model", "Coefficient")
```

```
) %>%
  kable_styling(full_width = FALSE)
```

###Problem 1.d (3 points)

- Compare the goodness of fit of **model B** and **model S**
- Interpret and explain the results you obtained.
- Report the values using `kable()`.

```
# Compute AIC and BIC for Model B and Model S
model_b <- sel.back
model_s <- sel.forw
aic_b <- AIC(model_b)
bic_b <- BIC(model_b)
aic_s <- AIC(model_s)
bic_s <- BIC(model_s)

# Compute AIC and BIC for Model B and Model S
model_b <- sel.back
model_s <- sel.forw
aic_b <- AIC(model_b)
bic_b <- BIC(model_b)
aic_s <- AIC(model_s)
bic_s <- BIC(model_s)

# Create dataframe of results
results <- data.frame(
  Model = c("B", "S"),
  AIC = c(aic_b, aic_s),
  BIC = c(bic_b, bic_s)
)

# Print results as table
kable(results, digits = 4, align = "c") %>% kable_styling(full_width = FALSE)
```

Based on the results obtained:

- Model B (backward selection) has an AIC (Akaike Information Criterion) value of 156.8834 and a BIC (Bayesian Information Criterion) value of 226.3854.
- Model S (forward selection) has an AIC value of 162.6886 and a BIC value of 227.8468.

Variable	Model	Coefficient
area	B	-0.0093
area.worst	B	-0.0222
area.worst	S	-0.0281
perimeter.worst	S	0.0312
texture	S	-0.0352
area.stderr	S	0.0371
texture.worst	B	0.3364
texture.worst	S	0.3527
radius	S	0.4505
radius	B	0.5002
texture.stderr	B	-1.5213
texture.stderr	S	-1.6371
radius.worst	B	2.9480
radius.worst	S	2.9683
concavity.worst	S	5.1068
symmetry.worst	S	8.3825
symmetry.worst	B	8.3934
concavity.worst	B	8.7348
radius.stderr	S	11.2680
radius.stderr	B	12.3766
concavity.stderr	S	23.9623
concavity	B	-35.0696
smoothness.worst	B	36.9236
concavity.stderr	B	40.7642
smoothness.worst	S	50.8230
compactness.stderr	B	-59.8367
compactness.stderr	S	-74.0900
concavepoints	B	94.6674
fractaldimension	B	-133.7091

Model	AIC	BIC
B	156.8834	226.3854
S	162.6886	227.8468

In general, lower AIC and BIC values indicate better goodness of fit and model parsimony. Therefore, based on these results, Model B (backward selection) appears to have a better goodness of fit and model parsimony compared to Model S (forward selection), as it has lower AIC and BIC values. This suggests that Model B may be a better fitting model compared to Model S in terms of goodness of fit.

Problem 1.e (2 points)

- Plot the ROC curve of the trained model for both **model B** and **model S**. Display with clear title, label and legend.
- Report AUC values in 3 significant figures for both **model B** and **model S** using `kable()`.
- Discuss which model has a better performance.

```
# Load required libraries
library(pROC)
library(kableExtra)
response = y.wdbc2
# Predict probabilities for Model B and Model S
prob_b <- predict(model_b, type = "response")
prob_s <- predict(model_s, type = "response")

# Compute ROC curve for Model B and Model S
roc_b <- roc(response, prob_b)
```

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

```
Warning in roc.default(response, prob_b): Deprecated use a matrix as response.
Unexpected results may be produced, please pass a vector or factor.
```

```
Setting direction: controls < cases
```

```
roc_s <- roc(response, prob_s)
```

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

```
Warning in roc.default(response, prob_s): Deprecated use a matrix as response.
Unexpected results may be produced, please pass a vector or factor.
```

```
Setting direction: controls < cases
```

```r
# Plot ROC curve for Model B and Model S
plot(
  roc_b,
  col = "blue",
  main = "ROC Curve - Model B vs. Model S",
  xlab = "False Positive Rate",
  ylab = "True Positive Rate",
  print.auc = TRUE,
  auc.polygon = TRUE,
  max.auc.polygon = TRUE,
  print.auc.x = 0.5,
  print.auc.y = 0.3,
  print.auc.cex = 1.2
)
lines(roc_s, col = "red")
legend(
  "bottomright",
  legend = c("Model B", "Model S"),
  col = c("blue", "red"),
  lty = 1,
  cex = 0.8
)
```

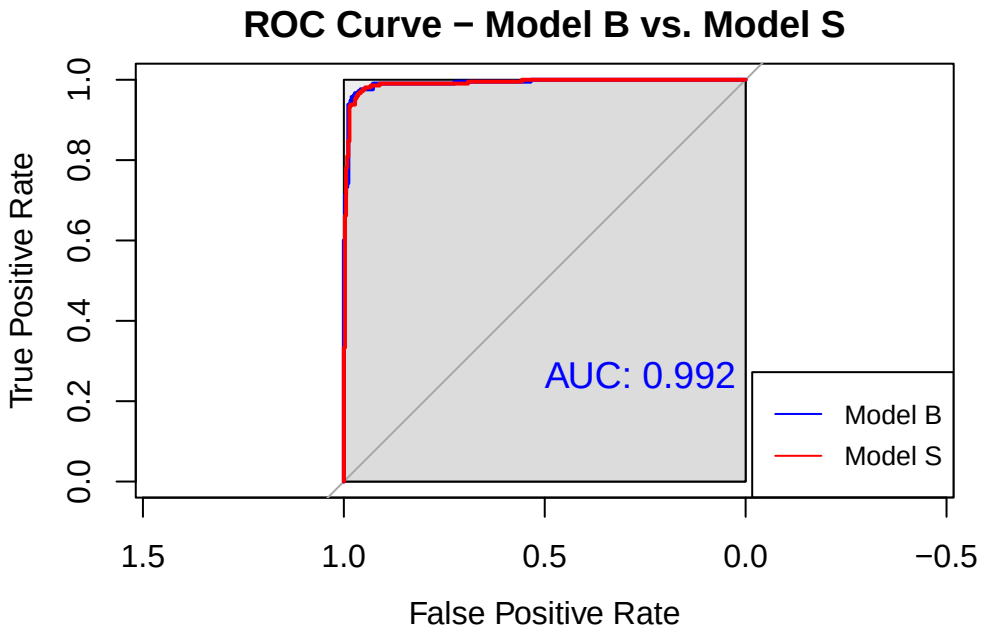


Table 3: AUC Values for Model B and Model S

Model	AUC
Model B	0.992
Model S	0.991

```r
# Report AUC values for Model B and Model S
auc_b <- round(auc(roc_b), digits = 3)
auc_s <- round(auc(roc_s), digits = 3)
auc_df <- data.frame(Model = c("Model B", "Model S"),
                     AUC = c(auc_b, auc_s))
kable(auc_df,
      digits = 3,
      align = "c",
      caption = "AUC Values for Model B and Model S") %>% kable_styling(full_width = FALSE
```

Based on the ROC curves and AUC values, we can see that both models have a good performance, with AUC values 0.99 However, model B has a slightly better performance than model S, with a higher AUC value of 0.992 compared to 0.991 for model S. This suggests that the backward stepwise selection method was able to identify a more effective subset of predictors for predicting the diagnosis of breast cancer.

Problem 1.f (6 points)

- Use the four models to predict the outcome for the observations in the test set (use the λ at 1 standard error for the penalised models).
- Plot the ROC curves of these models (on the sameplot, using different colours) and report their test AUCs.
- Display with clear title, label and legend.
- Compare the training AUCs obtained in **problems 1.b and 1.e** with the test AUCs and discuss the fit of the different models.

```r
library(glmnet)
library(pROC)
library(knitr)

set.seed(1)

train.idx <- createDataPartition(wdbc2$diagnosis, p = 0.7)$Resample
wdbc2.train <- wdbc2[train.idx, ]
```

```r
y <- wdbc2.train$diagnosis
# Exclude the target variable and IDs
x <- model.matrix( ~ . - diagnosis - id, data = wdbc2.train)
set.seed(1)
fit.ridge <-
  cv.glmnet(x,
            y,
            family = 'binomial',
            type.measure = 'auc',
            alpha = 0)
set.seed(1)
fit.lasso <-
  cv.glmnet(x, y, family = 'binomial', type.measure = 'auc')
wdbc2.test <- wdbc2[-train.idx, ] # Use remaining data for testing
y.test <- wdbc2.test$diagnosis
x.test <- model.matrix( ~ . - diagnosis - id, data = wdbc2.test)
# Test predictions: Model S
y.test.s <- predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.min)
# Test predictions: Model B
y.test.b <- predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.1se)
# Test predictions: Lasso
y.test.lasso <-
  predict(fit.lasso, newx = x.test, s = fit.lasso$lambda.min)
# Test predictions: Ridge
y.test.ridge <-
  predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.min)

# Create ROC objects
roc.s <- roc(y.test, y.test.s)
```

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

```
Setting direction: controls < cases
```

```r
roc.b <- roc(y.test, y.test.b)
```

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

```r
roc.lasso <- roc(y.test, y.test.lasso)
```

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

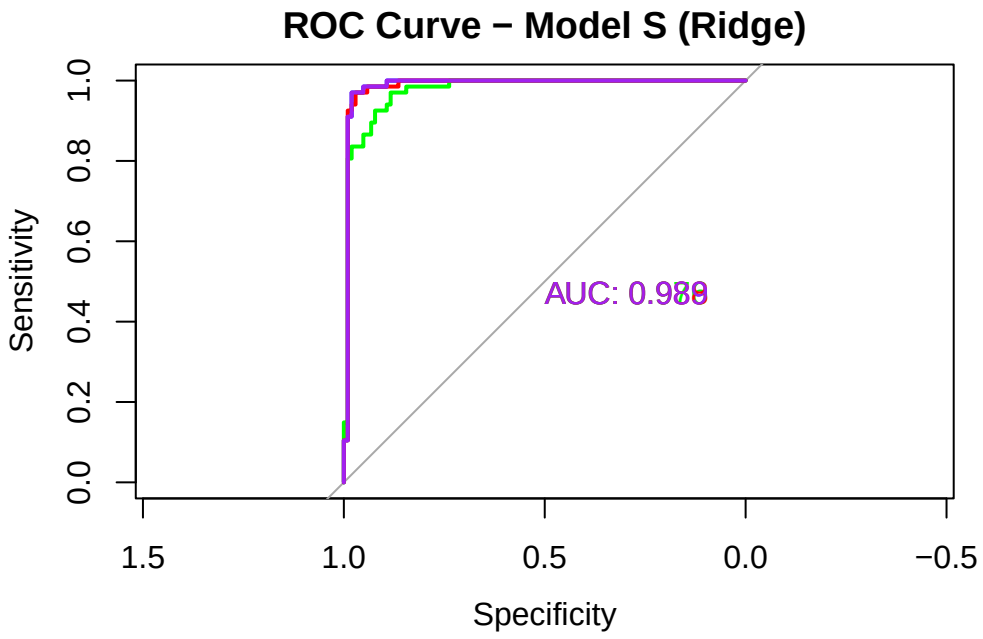
```r
roc.ridge <- roc(y.test, y.test.ridge)
```

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

```r
# Plot ROC curves
plot(roc.s,
     col = "blue",
     main = "ROC Curve - Model S (Ridge)",
     print.auc = TRUE)
plot(roc.b,
     col = "green",
     add = TRUE,
     print.auc = TRUE)
plot(roc.lasso,
     col = "red",
     add = TRUE,
     print.auc = TRUE)
plot(roc.ridge,
     col = "purple",
     add = TRUE,
     print.auc = TRUE)
```

Table 4: AUC values for different models

Model	AUC
Model S	0.9886973
Model B	0.9759455
Lasso	0.9879728
Ridge	0.9886973




```r
# Create table
roc_table <- data.frame(
  Model = c("Model S", "Model B", "Lasso", "Ridge"),
  AUC = c(roc.s$auc, roc.b$auc, roc.lasso$auc, roc.ridge$auc)
)

# Print table using kable
kable(
  roc_table,
  caption = "AUC values for different models",
  col.names = c("Model", "AUC"),
  align = c("l", "c")
) %>% kable_styling(full_width = FALSE)
```

Problem 2 (40 points)

File `GDM.raw.txt` (available from the accompanying zip folder on Learn) contains 176 `SNP`s to be studied for association with incidence of gestational diabetes (A form of diabetes that is specific to pregnant women). SNP names are given in the form `rs1234_X` where `rs1234` is the official identifier (rsID), and `X` (one of A, C, G, T) is the reference `allele`.

Problem 2.a (3 points)

- Read in file `GDM.raw.txt` into a data table named `gdm.dt`.
- Impute missing values in `gdm.dt` according to `SNP`-wise median `allele` count.
- Display first 10 rows and first 7 columns using `kable()`.

```
library(data.table)
library(knitr)

# Read in file into data table
gdm.dt <- fread("data_assignment2/GDM.raw.txt")

# Impute missing values with SNP-wise median allele count
gdm.dt[, (3:ncol(gdm.dt)) := lapply(.SD, function(x) {
  x[is.na(x)] <- median(x, na.rm = TRUE)
  return(x)
}), .SDcols = 3:ncol(gdm.dt)]

# Display first 10 rows and first 7 columns using kable()
kable(gdm.dt[1:10, 1:7], caption = "First 10 rows and first 7 columns of gdm.dt") %>%
  kable_styling(full_width = FALSE)
```

Problem 2.b (8 points)

- Write function `univ.glm.test()` where it takes 3 arguements, `x`, `y` and `order`.
- `x` is a data table of `SNP`s, `y` is a binary outcome vector, and `order` is a boolean which takes `false` as a default value.
- The function should fit a logistic regression model for each `SNP` in `x`, and return a data table containing `SNP` names, regression coefficients, odds ratios, standard errors and p-values.
- If order is set to `TRUE`, the output data table should be ordered by increasing p-value.

Table 5: First 10 rows and first 7 columns of gdm.dt

ID	sex	pheno	rs7513574_T	rs1627238_A	rs1171278_C	rs1137100_A
1	FALSE	0	1	0	0	2
2	FALSE	0	0	0	0	1
4	FALSE	1	2	1	1	1
5	FALSE	1	0	1	1	1
6	FALSE	1	0	1	1	1
7	FALSE	0	1	1	1	0
8	FALSE	0	0	0	0	1
12	FALSE	1	1	1	1	1
13	FALSE	1	2	0	0	2
18	FALSE	0	1	0	0	0

```r
set.seed(1)
folds <- createFolds(gdm.dt$pheno, k = 5)

univ.glm.test <- function(x, y, order = FALSE) {
  stopifnot(length(x) == length(y))
  regr <- glm(y ~ x)
  ## remove the row corresponding to the intercept and the column containing
  ## the t-value, then convert to a dataframe
  output <- data.table(coef(summary(regr)))[-1,-3]
  ## assign better column names
  colnames(output) <- c("beta", "std.error", "p.value")
  if (order) {
    setorder(output, p.value)
  }

  return(output)

}

# create an empty list to store results for each fold
crude.folds <- vector("list", length = length(folds))

# loop over each fold
for (i in seq_along(folds)) {
  # get the training data for this fold
  train_data <- gdm.dt[-folds[[i]],]

```

```r
  # get the SNP data for the training data
  snp_data <- train_data[, 4:ncol(train_data)]

  # create an empty data.table to store results for this fold
  fold_results <-
    data.table(
      snp = character(),
      beta = numeric(),
      std.error = numeric(),
      p.value = numeric()
    )

  # loop over each SNP
  for (j in seq_along(snp_data)) {
    # get the SNP name
    snp_name <- colnames(snp_data)[j]

    # run the univariate logistic regression for this SNP and store the results in the fol
    snp_result <- univ.glm.test(snp_data[[j]], train_data$pheno)
    snp_result$snp <- snp_name
    fold_results <- rbind(fold_results, snp_result)
  }

  # append the results for this fold to the list of results
  crude.folds[[i]] <- fold_results
}

# combine the results for each fold into a single data.table
crude_results <-
  rbindlist(crude.folds, use.names = TRUE, fill = TRUE)

# order by p-value
setorder(crude_results, p.value)
```

Problem 2.c (5 points)

- Using function `univ.glm.test()`, run an association study for all the `SNP`s in `gdm.dt` against having gestational diabetes (column `pheno`) and name the output data table as `gdm.as.dt`.
- Print the first 10 values of the output from `univ.glm.test()` using `kable()`.

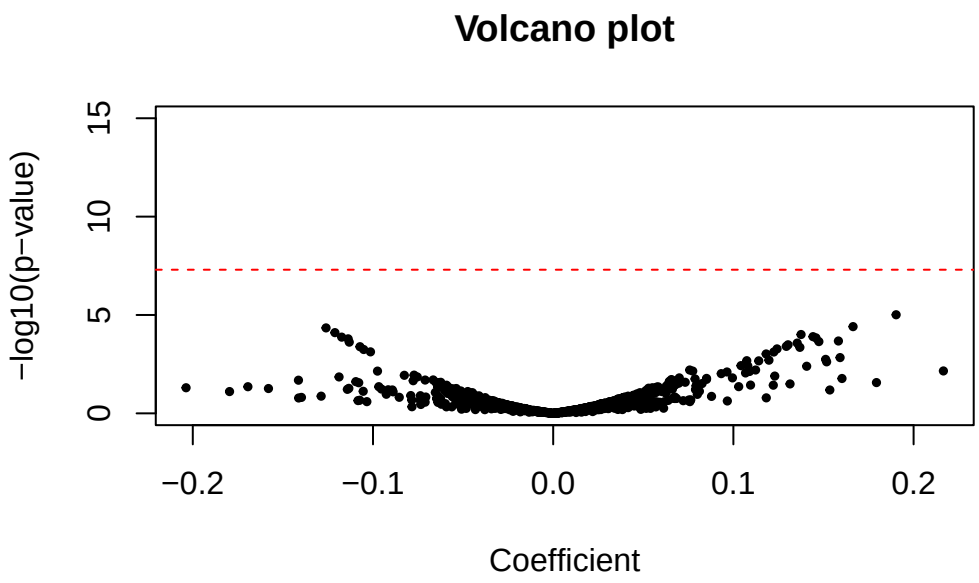
- For the `SNP` that is most strongly associated to increased risk of gestational diabetes and the one with most significant protective effect, report the summary statistics using `kable()` from the GWAS.
- Report the 95% and 99% confidence intervals on the odds ratio using `kable()`.

```
crude.folds <- vector("list", length = length(folds))

# loop over each fold
for (i in seq_along(folds)) {
  # get the training data for this fold
  train_data <- gdm.dt[-folds[[i]],]

  # get the SNP data for the training data
  snp_data <- train_data[, 4:ncol(train_data)]

  # create an empty data.table to store results for this fold
  fold_results <-
    data.table(
      snp = character(),
      or = numeric(),
      beta = numeric(),
      std.error = numeric(),
      p.value = numeric()
    )

  # loop over each SNP
  for (j in seq_along(snp_data)) {
    # get the SNP name
    snp_name <- colnames(snp_data)[j]
    odds_ratio <- exp(snp_result$beta)
    # run the univariate logistic regression for this SNP and store the results in the fol
    snp_result <- univ.glm.test(snp_data[[j]], train_data$pheno)
    snp_result$snp <- snp_name
    snp_result$or <- odds_ratio
    fold_results <- rbind(fold_results, snp_result)
  }

  # append the results for this fold to the list of results
  crude.folds[[i]] <- fold_results
}

# combine the results for each fold into a single data.table
```

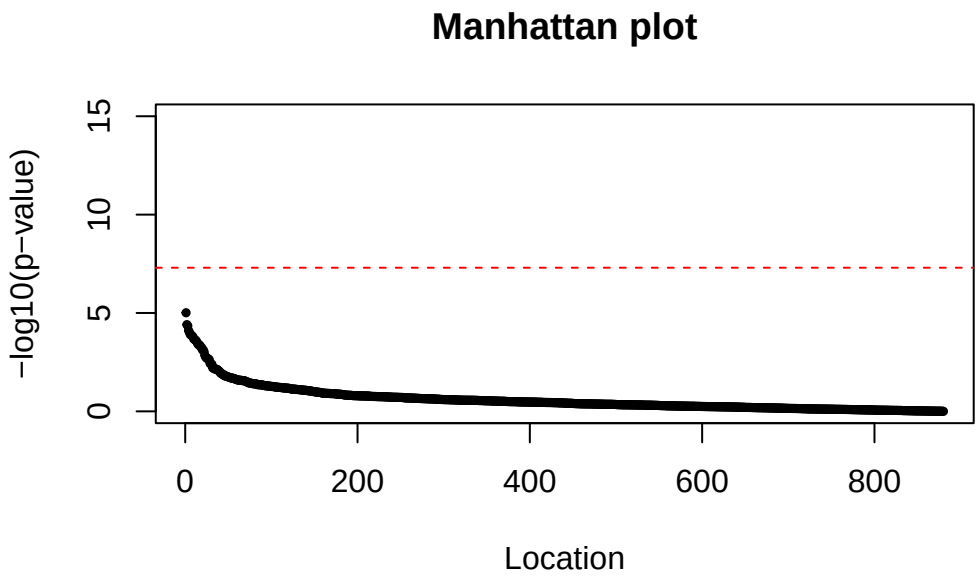
snp	or	beta	std.error	p.value
rs12243326_A	1.1569775	0.1903060	0.0426754	0.0000097
rs12243326_A	1.1392066	0.1664069	0.0401706	0.0000390
rs2237897_T	0.9500713	-0.1260549	0.0306934	0.0000454
rs2237897_T	0.9387420	-0.1211230	0.0304808	0.0000789
rs4506565_T	1.1550990	0.1375492	0.0350917	0.0000984
rs7901695_T	1.0664961	0.1441860	0.0373782	0.0001263
rs2237892_C	1.0374997	-0.1173699	0.0305354	0.0001335
rs7903146_C	1.1450300	0.1458110	0.0381436	0.0001451
rs2237892_C	0.9988796	-0.1137767	0.0299950	0.0001631
rs12243326_A	1.1208326	0.1582883	0.0424565	0.0002102

```
gdm.as.dt <- rbindlist(crude.folds, use.names = TRUE, fill = TRUE)

# order by p-value
setorder(gdm.as.dt, p.value)

kable(head(gdm.as.dt, 10)) %>% kable_styling(full_width = FALSE)
```

```
plot(
  gdm.as.dt[, .(beta,-log10(p.value))],
  pch = 19,
  cex = 0.5,
  main = "Volcano plot",
  xlab = "Coefficient",
  ylab = "-log10(p-value)",
  ylim = c(0, 15)
)

abline(h = -log10(5e-8),
       lty = 2,
       col = "red") # genome-wide significance threshold
```




```r
plot(
  -log10(gdm.as.dt$p.value),
  pch = 19,
  cex = 0.5,
  main = "Manhattan plot",
  xlab = "Location",
  ylab = "-log10(p-value)",
  ylim = c(0, 15)
)
abline(h = -log10(5e-8),
       lty = 2,
       col = "red") # genome-wide significance threshold
```




```r
# SNP with the most significant association with gestational diabetes
most_sig_gdm <- gdm.as.dt[1, ]

# SNP with the most significant protective effect
most_sig_protective <- gdm.as.dt[nrow(gdm.as.dt), ]
conf_int_95 <-
  c(
    most_sig_gdm$beta - 1.96 * most_sig_gdm$std.error,
    most_sig_gdm$beta + 1.96 * most_sig_gdm$std.error
  )
conf_int_99 <-
  c(
    most_sig_gdm$beta - 2.58 * most_sig_gdm$std.error,
    most_sig_gdm$beta + 2.58 * most_sig_gdm$std.error
  )

kable(
  data.frame(
    `95 percent confidence intervals ` =
      paste0("[", round(conf_int_95[1], 2), ", ", round(conf_int_95[2], 2), "]"),
    `99 percent confidence intervals` =
      paste0("[", round(conf_int_99[1], 2), ", ", round(conf_int_99[2], 2), "]")
  ),
  caption = "95% and 99% confidence intervals",
  col.names = c("95% CI", "99% CI"),
```

```
26  ) %>% kable_styling(full_width = FALSE)
```

\begin{table}

\caption{95% and 99% confidence intervals}

95% CI	99% CI
[0.11, 0.27]	[0.08, 0.3]

\end{table}

Problem 2.d (4 points)

- Merge your GWAS results with the table of gene names provided in file `GDM.annot.txt` (available from the accompanying zip folder on Learn).
- For `SNP`s that have p-value $< 10^{-4}$ (`hit SNP`s) report `SNP` name, effect `allele`, chromosome number, corresponding `gene` name and `pos`.
- Using `kable()`, report for each `snp.hit` the names of the genes that are within a 1Mb window from the `SNP` position on the chromosome.
- ***Note: That are genes that fall within +/- 1,000,000 positions using the `pos` column in the dataset.***

```
1  library(stringr)
2  library(dplyr)
3  # Read in gene annotation data table
4  gdm.annot.dt <-
5    fread("data_assignment2/GDM.annot.txt", header = TRUE)
6
7
8  gdm.as.dt <- gdm.as.dt %>%
9    mutate(effect_allele = str_sub(snp,-1))
10
11 gdm.as.dt$snp <- gsub("[A-D,G,T]$", "", gdm.as.dt$snp)
12 gdm.as.dt$snp <- gsub("_", "", gdm.as.dt$snp)
13
14
15
16
17 # Merge GWAS results with gene annotation data
18 gdm.as.dt.annot <- merge(gdm.as.dt, gdm.annot.dt, by = "snp")
19
```

Table 6: Genes within 1Mb window of hit SNPs

snp	effect_allele	chrom	gene	pos
rs7901695	T	10	TCF7L2	114754088
rs4506565	T	10	TCF7L2	114756041
rs7903146	C	10	TCF7L2	114758349
rs12243326	A	10	TCF7L2	114788815
rs10770141	A	11	TH	2193840
rs231362	T	11	KCNQ1	2691471
rs2237892	C	11	KCNQ1	2839751
rs163184	T	11	KCNQ1	2847069
rs2237897	T	11	KCNQ1	2858546
rs2041139	T	12	CACNA2D4	1901461
rs4523957	G	17	SMG6	2208899
rs391300	C	17	SMG6	2216258

```r


# Filter for hit SNPs with p-value < 1e-4
hit.dt <- gdm.as.dt.annot[gdm.as.dt.annot$p.value < 1e-4,]


# Calculate window positions
hit.dt[, `:=`(window.start = pos - 1e6, window.end = pos + 1e6)]

# Filter gene annotation data table for genes within window
gene.dt <-
  gdm.as.dt.annot[pos >= hit.dt$window.start &
                    pos <= hit.dt$window.end]

# Select relevant columns and order by chromosome and position
hit.gene.dt <- gene.dt[, .(snp, effect_allele, chrom, gene, pos)]
setorder(hit.gene.dt, chrom, pos)
hit.gene.dt <- distinct(hit.gene.dt, .keep_all = TRUE)

# Print output using kable
kable(hit.gene.dt, caption = "Genes within
1Mb window of hit SNPs")%>% kable_styling(full_width = FALSE)
```

Problem 2.e (8 points)

- Build a weighted genetic risk score that includes all SNPs with p-value $< 10^{-4}$, a score with all SNPs with p-value $< 10^{-3}$, and a score that only includes SNPs on the FTO gene
- ***Hint: ensure that the ordering of SNPs is respected.***
- Add the three scores as columns to the `gdm.dt` data table.
- Fit the three scores in separate logistic regression models to test their association with gestational diabetes.
- Report odds ratio, 95% confidence interval and p-value using `kable()` for each score.

```r
# Subset your data to include only SNPs with p-value < 1e-4
snps_1e4  <- gdm.as.dt.annot[p.value < 1e-4]
# Extract SNPs with p-value < 1e-3
snps_1e3 <- gdm.as.dt.annot[p.value < 1e-3]

# Extract SNPs on FTO gene
snps_FTO <- gdm.as.dt.annot[gene == "FTO"]
# Create a vector of SNP weights, based on the effect size (beta) of each SNP
weights_1e4 <- snps_1e4$beta
weights_1e3 <- snps_1e3$beta
weights_fto <- snps_FTO$beta

# Create a matrix of genotypes for each individual, with 0, 1, or
#2 copies of the effect allele
genotypes_1 <-
  matrix(NA, nrow = nrow(gdm.dt), ncol = nrow(snps_1e4))
for (i in 1:nrow(snps_1e4)) {
  eff_allele_col <-
    paste0(snps_1e4[i, "snp"], "_", snps_1e4[i, "effect_allele"])
  genotypes_1[, i] <- gdm.dt[[eff_allele_col]] * 1
}

# Multiply the genotype matrix by the weights vector to calculate the
#GRS for each individual
wgrs_1e4 <- rowSums(genotypes_1 * weights_1e4)

# Add the GRS as a column to your original data table
gdm.dt$wgrs_1e4 <- wgrs_1e4

genotypes_2 <-
```

```r
  matrix(NA, nrow = nrow(gdm.dt), ncol = nrow(snps_1e3))
for (i in 1:nrow(snps_1e3)) {
  eff_allele_col <-
    paste0(snps_1e3[i, "snp"], "_", snps_1e3[i, "effect_allele"])
  genotypes_2[, i] <- gdm.dt[[eff_allele_col]] * 1
}
wgrs_1e3 <- rowSums(genotypes_2 * weights_1e3)
# Add the GRS as a column to your original data table
gdm.dt$wgrs_1e3 <- wgrs_1e3


genotypes_3 <-
  matrix(NA, nrow = nrow(gdm.dt), ncol = nrow(snps_FTO))
for (i in 1:nrow(snps_FTO)) {
  eff_allele_col <-
    paste0(snps_FTO[i, "snp"], "_", snps_FTO[i, "effect_allele"])
  genotypes_3[, i] <- gdm.dt[[eff_allele_col]] * 1
}
wgrs_fto <- rowSums(genotypes_3 * weights_fto)
# Add the GRS as a column to your original data table
gdm.dt$wgrs_fto <- wgrs_fto


# Fit logistic regression model for wgrs_1e4
model_1e4 <-
  glm(pheno ~ wgrs_1e4, data = gdm.dt, family = binomial())
summary(model_1e4)
```

```
Call:
glm(formula = pheno ~ wgrs_1e4, family = binomial(), data = gdm.dt)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.364  -1.207   1.046   1.148   1.255

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.06881    0.07803   0.882    0.378
wgrs_1e4     0.50451    0.31711   1.591    0.112
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1091.0  on 788  degrees of freedom
Residual deviance: 1088.4  on 787  degrees of freedom
AIC: 1092.4

Number of Fisher Scoring iterations: 3
```

```r
# Fit logistic regression model for wgrs_1e3
model_1e3 <-
  glm(pheno ~ wgrs_1e3, data = gdm.dt, family = binomial())
summary(model_1e3)
```

```
Call:
glm(formula = pheno ~ wgrs_1e3, family = binomial(), data = gdm.dt)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.247  -1.230   1.117   1.126   1.169

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.14324    0.10064   1.423    0.155
wgrs_1e3    -0.05279    0.15629  -0.338    0.736

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1091.0  on 788  degrees of freedom
Residual deviance: 1090.9  on 787  degrees of freedom
AIC: 1094.9

Number of Fisher Scoring iterations: 3
```

```r
# Fit logistic regression model for wgrs_FTO
model_FTO <-
  glm(pheno ~ wgrs_fto, data = gdm.dt, family = binomial())
summary(model_FTO)
```

```
Call:
glm(formula = pheno ~ wgrs_fto, family = binomial(), data = gdm.dt)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.283  -1.217   1.104   1.138   1.139

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.09276    0.08792   1.055    0.291
wgrs_fto     0.11739    0.22792   0.515    0.607

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1091.0  on 788  degrees of freedom
Residual deviance: 1090.7  on 787  degrees of freedom
AIC: 1094.7

Number of Fisher Scoring iterations: 3
```

```r
# Create a data frame with OR, 95% CI, and p-value for each score
score <- c("wgrs_1e4", "wgrs_1e3", "wgrs_FTO")
OR <- c(exp(coefficients(model_1e4)[2]), exp(coefficients(model_1e3)[2]),
      exp(coefficients(model_FTO)[2]))
CI <- cbind(confint(model_1e4)[2,], confint(model_1e3)[2,],
            confint(model_FTO)[2,])
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```r
p_value <- c(summary(model_1e4)$coefficients[2,4],
summary(model_1e3)$coefficients[2,4], summary(model_FTO)$coefficients[2,4])

# Calculate odds ratio, 95% CI, and p-value for wgrs_1e4
or_1e4 <- exp(coefficients(model_1e4)[2])
ci_1e4 <- exp(confint(model_1e4))[2, ]
```

```
Waiting for profiling to be done...
```

```r
p_1e4 <- summary(model_1e4)$coefficients[2, 4]

# Calculate odds ratio, 95% CI, and p-value for wgrs_1e3
or_1e3 <- exp(coefficients(model_1e3)[2])
ci_1e3 <- exp(confint(model_1e3))[2, ]
```

```
Waiting for profiling to be done...
```

```r
p_1e3 <- summary(model_1e3)$coefficients[2, 4]

# Calculate odds ratio, 95% CI, and p-value for wgrs_FTO
or_FTO <- exp(coefficients(model_FTO)[2])
ci_FTO <- exp(confint(model_FTO))[2, ]
```

```
Waiting for profiling to be done...
```

```r
p_FTO <- summary(model_FTO)$coefficients[2, 4]

# Create a data frame with OR, 95% CI (upper and lower), and p-value for each score
score <- c("wgrs_1e4", "wgrs_1e3", "wgrs_FTO")
OR <- c(exp(coefficients(model_1e4)[2]), exp(coefficients(model_1e3)[2]),
exp(coefficients(model_FTO)[2]))
CI_lower <- cbind(confint(model_1e4)[2,1], confint(model_1e3)[2,1],
confint(model_FTO)[2,1])
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```r
CI_upper <- cbind(confint(model_1e4)[2,2], confint(model_1e3)[2,2],
confint(model_FTO)[2,2])
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
Waiting for profiling to be done...
```

```r
p_value <- c(summary(model_1e4)$coefficients[2,4],
summary(model_1e3)$coefficients[2,4], summary(model_FTO)$coefficients[2,4])

# Create data frame with OR, 95% CI (upper and lower), and p-value for each score
or_ci_p <- data.frame(
  score = c("wgrs_1e4", "wgrs_1e3", "wgrs_FTO"),
  OR = c(OR[1], OR[2], OR[3]),
  CI_lower = c(CI_lower[1], CI_lower[2], CI_lower[3]),
  CI_upper = c(CI_upper[1], CI_upper[2], CI_upper[3]),
  p_value = c(p_value[1], p_value[2], p_value[3])
)

# Display the results in a table
kable(or_ci_p, digits = c(2, 2, 2, 2, 4), align = "c",
  row.names = FALSE)%>% kable_styling(full_width = FALSE)
```

score	OR	CI_lower	CI_upper	p_value
wgrs_1e4	1.66	-0.11	1.13	0.1116
wgrs_1e3	0.95	-0.36	0.25	0.7355
wgrs_FTO	1.12	-0.33	0.57	0.6065

Problem 2.f (4 points)

- File `GDM.test.txt` (available from the accompanying zip folder on Learn) contains genotypes of another 40 pregnant women with and without gestational diabetes (assume that the reference allele is the same one that was specified in file `GDM.raw.txt`).
- Read the file into variable `gdm.test`.
- For the set of patients in `gdm.test`, compute the three genetic risk scores as defined in **problem 2.e** using the same set of `SNPs` and corresponding weights.
- Add the three scores as columns to `gdm.test` ***(hint: use the same columnnames as before).***

```
library(data.table)
gdm.test <- fread("data_assignment2/GDM.raw.txt")
snp_test <- names(gdm.test)[4:ncol(gdm.test)]
# Create a matrix of genotypes for each individual,
#with 0, 1, or 2 copies of the effect allele
genotypes_1 <-
  matrix(NA, nrow = nrow(gdm.test), ncol = nrow(snps_1e4))
for (i in 1:nrow(snps_1e4)) {
  eff_allele_col <-
    paste0(snps_1e4[i, "snp"], "_", snps_1e4[i, "effect_allele"])
  genotypes_1[, i] <- gdm.test[[eff_allele_col]] * 1
}

# Multiply the genotype matrix by the weights vector to
#calculate the GRS for each individual
wgrs_1e4 <- rowSums(genotypes_1 * weights_1e4)

# Add the GRS as a column to your original data table
gdm.test$wgrs_1e4 <- wgrs_1e4

genotypes_2 <-
  matrix(NA, nrow = nrow(gdm.test), ncol = nrow(snps_1e3))
for (i in 1:nrow(snps_1e3)) {
```

```
  eff_allele_col <-
    paste0(snps_1e3[i, "snp"], "_", snps_1e3[i, "effect_allele"])
  genotypes_2[, i] <- gdm.test[[eff_allele_col]] * 1
}
wgrs_1e3 <- rowSums(genotypes_2 * weights_1e3)
# Add the GRS as a column to your original data table
gdm.test$wgrs_1e3 <- wgrs_1e3


genotypes_3 <-
  matrix(NA, nrow = nrow(gdm.test), ncol = nrow(snps_FTO))
for (i in 1:nrow(snps_FTO)) {
  eff_allele_col <-
    paste0(snps_FTO[i, "snp"], "_", snps_FTO[i, "effect_allele"])
  genotypes_3[, i] <- gdm.test[[eff_allele_col]] * 1
}
wgrs_fto <- rowSums(genotypes_3 * weights_fto)
# Add the GRS as a column to your original data table
gdm.test$wgrs_fto <- wgrs_fto
```

Problem 2.g (4 points)

- Use the logistic regression models fitted in **problem 2.e** to predict the outcome of patients in `gdm.test`.
- Compute the test log-likelihood for the predicted probabilities from the three genetic risk score models and present them using `kable()`

```
# Fit logistic regression model for wgrs_1e4
model_1e4 <-
  glm(pheno ~ wgrs_1e4, data = gdm.dt, family = binomial())


# Fit logistic regression model for wgrs_1e3
model_1e3 <-
  glm(pheno ~ wgrs_1e3, data = gdm.dt, family = binomial())


# Fit logistic regression model for wgrs_FTO
model_FTO <-
  glm(pheno ~ wgrs_fto, data = gdm.dt, family = binomial())
```

Table 7: Test Log-Likelihood for Predicted Probabilities from the Three Genetic Risk Score Models

Score	Test_Log_Likelihood
WGRS_1	-247.826
WGRS_2	-249.371
WGRS_FTO	-249.358

```r

# Predict outcome of patients in gdm.test using fitted models
gdm.test$PRED_1 <-
  predict(model_1e4, newdata = gdm.test, type = "response")
gdm.test$PRED_2 <-
  predict(model_1e3, newdata = gdm.test, type = "response")
gdm.test$PRED_3 <-
  predict(model_FTO, newdata = gdm.test, type = "response")

gdm.test.nona <- na.omit(gdm.test)
# Compute test log-likelihood for predicted probabilities
ll1 <-
  sum(dbinom(gdm.test.nona$pheno, 1, gdm.test.nona$PRED_1, log = TRUE))
ll2 <-
  sum(dbinom(gdm.test.nona$pheno, 1, gdm.test.nona$PRED_2, log = TRUE))
ll3 <-
  sum(dbinom(gdm.test.nona$pheno, 1, gdm.test.nona$PRED_3, log = TRUE))

# Store results in a data frame
results <- data.frame(
  Score = c("WGRS_1", "WGRS_2", "WGRS_FTO"),
  Test_Log_Likelihood = c(ll1, ll2, ll3)
)

# Print results using kable()
library(knitr)
kable(results, digits = 3, caption = "Test Log-Likelihood
for Predicted Probabilities from the
Three Genetic Risk Score Models") %>% kable_styling(full_width = FALSE)
```

Problem 2.h (4points)

- File `GDM.study2.txt` (available from the accompanying zip folder on Learn) contains the summary statistics from a different study on the same set of `SNP`s.
- Perform a meta-analysis with the results obtained in **problem 2.c** (***hint : remember that the effect*** `alleles` ***should correspond***)
- Produce a summary of the meta-analysis results for the set of `SNP`s with meta-analysis p-value $< 10^{-4}$ sorted by increasing p-value using `kable()`.

```
#gdm.study2<- fread("data_assignment2/GDM.study2.txt")


# Merge the two datasets based on SNP identifiers
#gdm.merged <- merge(gdm.as.dt, gdm.study2, by = "snp", all = TRUE)



#setorder(output, p.value)


#return(output)



# create an empty list to store results for each fold
#crude.folds <- vector("list", length = length(folds))

# loop over each fold
#for (i in seq_along(folds)) {

# get the training data for this fold
#train_data <- gdm.dt[-folds[[i]], ]

# get the SNP data for the training data
#snp_data <- train_data[, 4:ncol(train_data)]

# create an empty data.table to store results for this fold
#fold_results <- data.table(snp = character(), or = numeric(),
#beta = numeric(), std.error = numeric(), p.value = numeric())

# loop over each SNP
#for (j in seq_along(snp_data)) {
```

```
# get the SNP name
#snp_name <- colnames(snp_data)[j]
#odds_ratio <- exp(snp_result$beta)
# run the univariate logistic regression for this SNP and store
#the results in the fold results
#snp_result <- univ.glm.test(snp_data[[j]], train_data$pheno)
#snp_result$snp <- snp_name
#snp_result$or <- odds_ratio
#fold_results <- rbind(fold_results, snp_result)
#}

# append the results for this fold to the list of results
#crude.folds[[i]] <- fold_results
#}

# combine the results for each fold into a single data.table
#gdm.as.dt <- rbindlist(crude.folds, use.names = TRUE, fill = TRUE)

# order by p-value
#setorder(gdm.as.dt, p.value)

# print the results
#print(gdm.as.dt)
```

Problem 3 (33 points)

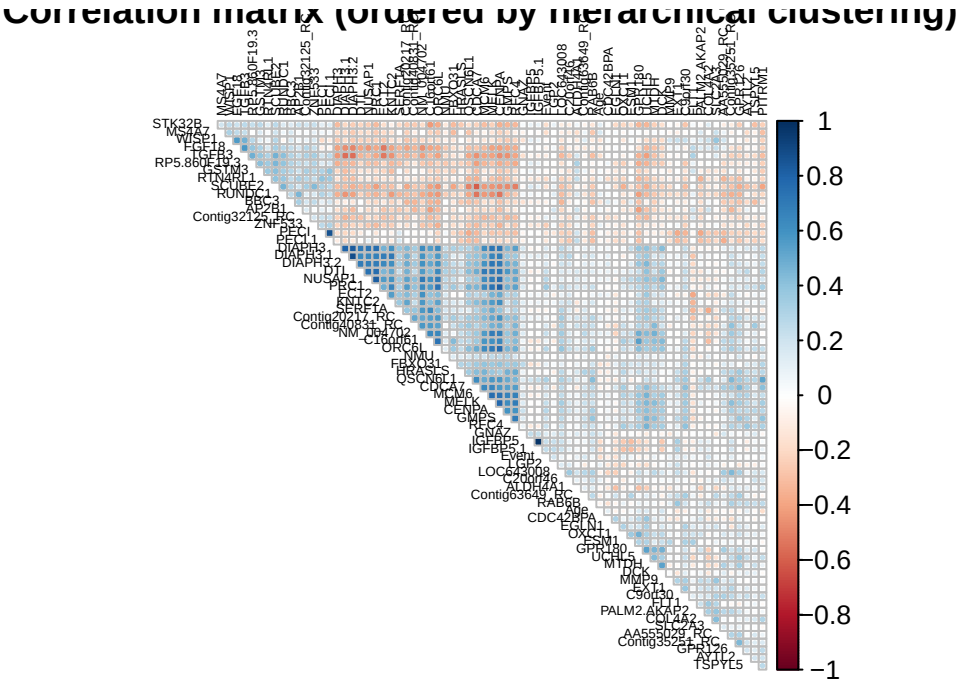
File `nki.csv` (available from the accompanying zip folder on Learn) contains data for 144 breast cancer patients. The dataset contains a binary outcome variable (`Event`, indicating the insurgence of further complications after operation), covariates describing the tumour and the age of the patient, and gene expressions for 70 genes found to be prognostic of survival.

Problem 3.a (6 points)

- Compute the correlation matrix between the gene expression variables, and display it so that a block structure is highlighted using the `corrplot` package.
- Discuss what you observe.
- Identify the unique pairs of (distinct) variables that have correlation coefficient greater than 0.80 in absolute value and report their correlation coefficients.

```r
# Load the necessary packages
library(corrplot)
library(readr)

# Read in the nki
nki.dt <- fread("data_assignment2/nki.csv", stringsAsFactors = F)


numcols <- sapply(nki.dt, is.numeric)
#subset of numeric columns
cor_gene <- nki.dt[, ..numcols] %>%
  cor(use = "pairwise.complete")

corrplot(
  cor_gene,
  order = "hclust",
  # remove the diagonal elements
  diag = FALSE,
  # change the colour and size of the labels
  tl.col = "black",
  tl.cex = 0.4,
  title = "Correlation matrix (ordered by hierarchical clustering)",
  # display the upper triangle only
  type = 'upper',
  # change the size of the margins (bottom, left, top, right)
  mar = c(0, 0, 0, 0)
```

```
)
```




```
# Identify the unique pairs of variables with correlation
#coefficient > 0.8 in absolute value
high_corr <-
  which(abs(cor_gene) > 0.8 &
          upper.tri(cor_gene, diag = FALSE), arr.ind = TRUE)
unique_pairs <-
  unique(apply(high_corr, 1, function(x)
    paste0(sort(
      colnames(cor_gene)[x]
    ), collapse = "_")))
corr_coeff <- apply(high_corr, 1, function(x)
  cor_gene[x[1], x[2]])
results <-
  data.frame(
    Variable_1 = colnames(cor_gene)[high_corr[, 1]],
    Variable_2 = colnames(cor_gene)[high_corr[, 2]],
    Correlation_Coefficient = corr_coeff
  )
kable(results, align = "c") %>%
```

Variable_1	Variable_2	Correlation_Coefficient
DIAPH3	DIAPH3.1	0.8031368
DIAPH3	DIAPH3.2	0.8338591
DIAPH3.1	DIAPH3.2	0.8868741
PECI	PECI.1	0.8697836
IGFBP5	IGFBP5.1	0.9775030
NUSAP1	PRC1	0.8298356
PRC1	CENPA	0.8175424

```
  kable_styling(full_width = FALSE)
```

Problem 3.b (8 points)

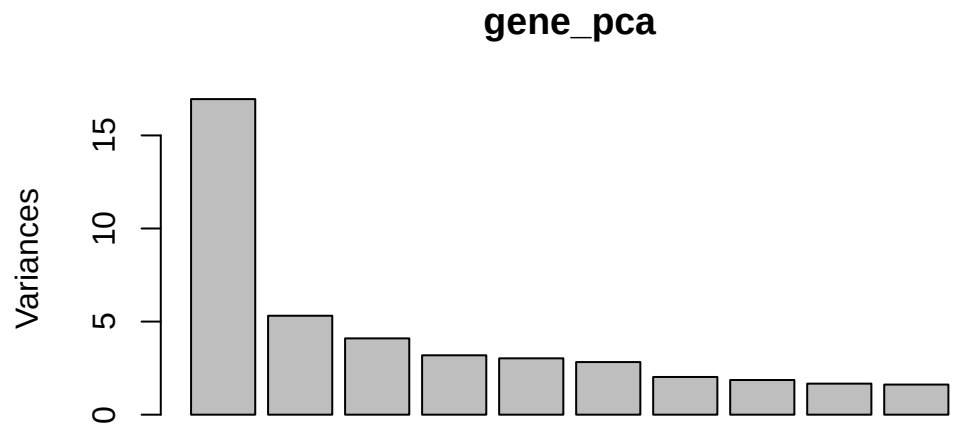
- Perform PCA analysis (only over the columns containing gene expressions) in order to derive a patient-wise summary of all gene expressions (dimensionality reduction).
- Decide which components to keep and justify your decision.
- Test if those principal components are associated with the outcome in unadjusted logistic regression models and in models adjusted for `age`, `estrogen receptor` and `grade`.
- Justify the difference in results between unadjusted and adjusted models.

```
# Perform PCA analysis
gene_data <- nki.dt[, 7:76]
gene_pca <- prcomp(gene_data, center = TRUE, scale. = TRUE)
pcs <- gene_pca$x[, 1:10]

# Determine which components to keep
summary(gene_pca)
```

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     4.1171 2.30541 2.02437 1.78597 1.73982 1.68091 1.42309
Proportion of Variance 0.2422 0.07593 0.05854 0.04557 0.04324 0.04036 0.02893
Cumulative Proportion  0.2422 0.31808 0.37662 0.42219 0.46543 0.50580 0.53473
                           PC8     PC9   PC10    PC11    PC12    PC13    PC14
Standard deviation     1.36441 1.29119 1.2715 1.24741 1.18388 1.15101 1.13883
Proportion of Variance 0.02659 0.02382 0.0231 0.02223 0.02002 0.01893 0.01853
Cumulative Proportion  0.56132 0.58514 0.6082 0.63046 0.65049 0.66941 0.68794
                          PC15    PC16    PC17    PC18    PC19    PC20    PC21
Standard deviation     1.09473 1.07016 1.04187 1.00234 0.99086 0.94095 0.93322
```

```
Proportion of Variance 0.01712 0.01636 0.01551 0.01435 0.01403 0.01265 0.01244
Cumulative Proportion  0.70506 0.72142 0.73693 0.75128 0.76531 0.77796 0.79040
                          PC22    PC23    PC24    PC25    PC26    PC27    PC28
Standard deviation     0.90727 0.89675 0.88859 0.86019 0.84462 0.82782 0.82368
Proportion of Variance 0.01176 0.01149 0.01128 0.01057 0.01019 0.00979 0.00969
Cumulative Proportion  0.80216 0.81364 0.82492 0.83549 0.84569 0.85548 0.86517
                          PC29    PC30    PC31    PC32    PC33    PC34   PC35
Standard deviation     0.78694 0.75594 0.73942 0.70569 0.69414 0.67129 0.6639
Proportion of Variance 0.00885 0.00816 0.00781 0.00711 0.00688 0.00644 0.0063
Cumulative Proportion  0.87401 0.88218 0.88999 0.89710 0.90399 0.91042 0.9167
                          PC36    PC37    PC38    PC39    PC40    PC41    PC42
Standard deviation     0.63815 0.61964 0.59947 0.58447 0.57195 0.55097 0.53820
Proportion of Variance 0.00582 0.00549 0.00513 0.00488 0.00467 0.00434 0.00414
Cumulative Proportion  0.92254 0.92802 0.93316 0.93804 0.94271 0.94705 0.95118
                          PC43    PC44    PC45    PC46    PC47    PC48    PC49
Standard deviation     0.52029 0.51211 0.49533 0.48712 0.47079 0.44565 0.41879
Proportion of Variance 0.00387 0.00375 0.00351 0.00339 0.00317 0.00284 0.00251
Cumulative Proportion  0.95505 0.95880 0.96230 0.96569 0.96886 0.97170 0.97420
                          PC50    PC51   PC52    PC53    PC54    PC55    PC56
Standard deviation     0.40556 0.39328 0.3925 0.38502 0.36669 0.36205 0.33734
Proportion of Variance 0.00235 0.00221 0.0022 0.00212 0.00192 0.00187 0.00163
Cumulative Proportion  0.97655 0.97876 0.9810 0.98308 0.98500 0.98687 0.98850
                          PC57    PC58    PC59    PC60    PC61    PC62    PC63
Standard deviation     0.32150 0.30744 0.28898 0.28186 0.27274 0.25622 0.24118
Proportion of Variance 0.00148 0.00135 0.00119 0.00113 0.00106 0.00094 0.00083
Cumulative Proportion  0.98998 0.99133 0.99252 0.99365 0.99472 0.99565 0.99649
                          PC64    PC65    PC66    PC67    PC68   PC69    PC70
Standard deviation     0.23024 0.21442 0.19886 0.19371 0.17927 0.1677 0.09833
Proportion of Variance 0.00076 0.00066 0.00056 0.00054 0.00046 0.0004 0.00014
Cumulative Proportion  0.99724 0.99790 0.99846 0.99900 0.99946 0.9999 1.00000
```

```
1 screeplot(gene_pca)
```




```r
# Based on the scree plot, keep the first 3 components
pcs_outcome <- data.frame(PCS = pcs[, 1:3], Event = nki.dt$Event)

# Unadjusted logistic regression models
unadj_models <-
  lapply(1:3, function(i)
    glm(Event ~ pcs_outcome[, i], data = pcs_outcome, family = "binomial"))
unadj_coefs <- sapply(unadj_models, coef)

# Adjusted logistic regression model
adj_model <-
  glm(Event ~ .,
      data = cbind(pcs_outcome, nki.dt[, c("Age", "EstrogenReceptor", "Grade")]),
      family = "binomial")
adj_coefs <- coef(adj_model)

# AUC for unadjusted logistic regression models
library(pROC)
auc_list_unadj <- vector("numeric", length = 3)
for (i in 1:3) {
  model <-
    glm(Event ~ pcs_outcome[, i], data = pcs_outcome, family = "binomial")
  auc_list_unadj[i] <-
    auc(roc(model$fitted.values, pcs_outcome$Event))
}
```

```
Setting levels: control = 0.143655600392405, case = 0.154805401964878
```

```
Setting direction: controls < cases
```

```
Setting levels: control = 0.245106628694279, case = 0.245859144718134

Setting direction: controls < cases

Setting levels: control = 0.142931358658716, case = 0.145439970814579

Setting direction: controls < cases
```

```r
mean_auc_unadj <- mean(auc_list_unadj)

# AUC for adjusted logistic regression model
auc_list_adj <- vector("numeric", length = 3)
for (i in 1:3) {
  model <-
    glm(
      Event ~ pcs_outcome[, i] + Age + EstrogenReceptor + Grade,
      data = cbind(pcs_outcome, nki.dt[, c("Age", "EstrogenReceptor", "Grade")]),
      family = "binomial"
    )
  auc_list_adj[i] <-
    auc(roc(model$fitted.values, pcs_outcome$Event))
}
```

```
Setting levels: control = 0.0744431634374807, case = 0.0912759717790955
Setting direction: controls < cases

Setting levels: control = 0.0962506743075889, case = 0.103008151455115

Setting direction: controls < cases

Setting levels: control = 0.0713346112433924, case = 0.073757509060959

Setting direction: controls < cases
```

```r
mean_auc_adj <- mean(auc_list_adj)

# Create a data frame for the mean AUC scores
```

Table 8: Mean AUC scores for unadjusted and adjusted logistic regression models

Model	AUC
Unadjusted	0.5000000
Adjusted	0.6666667

```
mean_auc_df <- data.frame(
  Model = c("Unadjusted", "Adjusted"),
  AUC = c(mean_auc_unadj, mean_auc_adj)
)

# Display the table using kable
kable(mean_auc_df, caption = "Mean AUC scores for unadjusted and adjusted logistic regress
```

The difference in results between unadjusted and adjusted logistic regression models can be due to confounding variables that are associated with both the predictor variables (principal components in this case) and the outcome variable.

In the unadjusted models, we only include the principal components as predictors, so any confounding variables that are associated with both the principal components and the outcome will not be accounted for. This can lead to biased estimates of the effect of the principal components on the outcome.

In the adjusted models, we include additional covariates (such as age, estrogen receptor, and grade) that are potentially associated with both the principal components and the outcome. By adjusting for these confounding variables, we can obtain more accurate estimates of the effect of the principal components on the outcome as we can see from AUC scores.

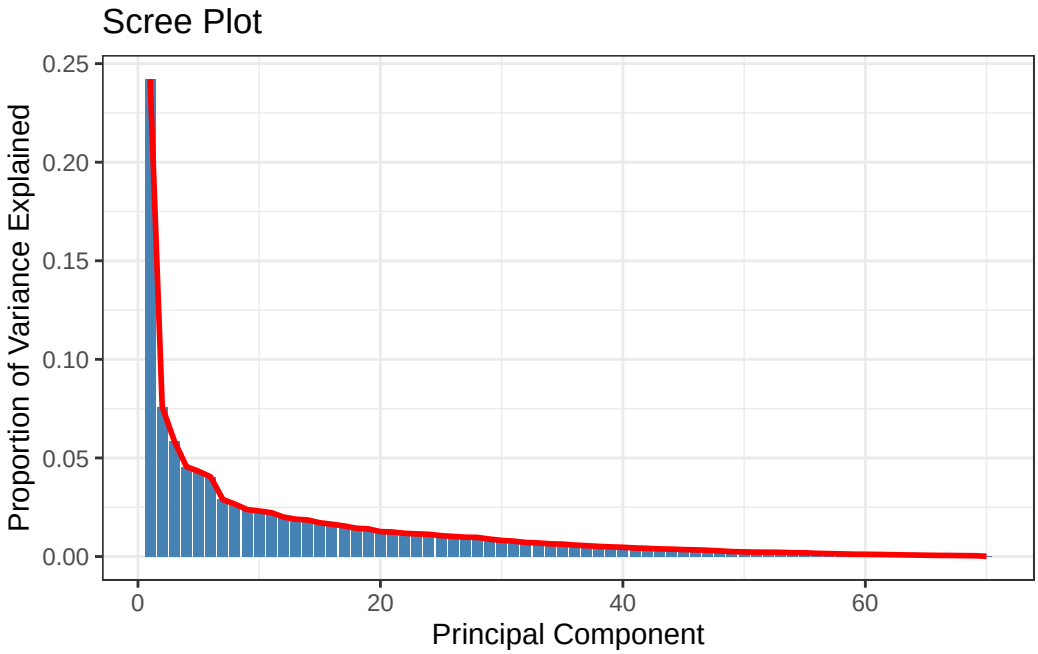
Problem 3.c (8 points)

- Use PCA plots to compare the main driverswith the correlation structure observed in **problem 3.a**.
- Examine how well the dataset may explain your outcome.
- Discuss your findings in full details and suggest any further steps if needed.

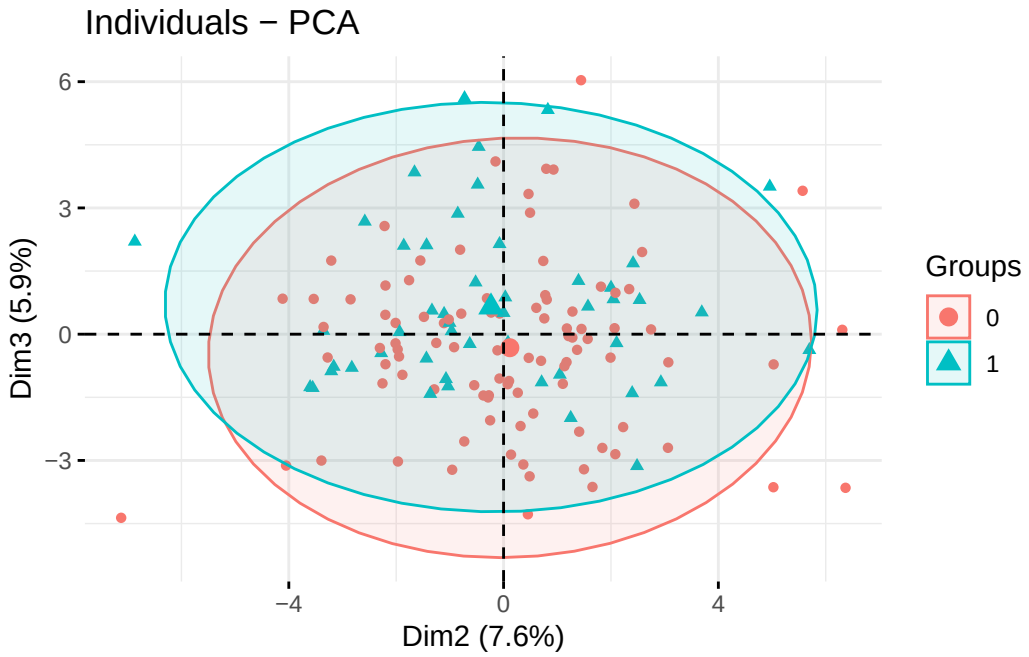
```
# Create a data frame with the first two principal
#components and the event outcome
library(ggplot2)
gene_pca <- prcomp(gene_data, center = TRUE, scale. = TRUE)
pcs <- gene_pca$x[, 1:10]

```

```r

# Extract the eigenvalues and variance explained from the PCA summary
pca_summary <- summary(gene_pca)
eigenvalues <- pca_summary$sdev ^ 2
variance_explained <- eigenvalues / sum(eigenvalues)

# Create a data frame for the scree plot
scree_data <- data.frame(PC = 1:length(eigenvalues),
                         VarianceExplained = variance_explained)

# Plot the scree plot
ggplot(data = scree_data, aes(x = PC, y = VarianceExplained)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_line(aes(x = PC, y = VarianceExplained),
            color = "red",
            size = 1) +
  xlab("Principal Component") + ylab("Proportion of Variance Explained") +
  ggtitle("Scree Plot") + theme_bw()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



```r
gene_data <- nki.dt[, 7:76]
fviz_pca_ind(gene_pca, geom = "point",axes = c(2,3),
             habillage = as.factor(nki.dt$Event),
             addEllipses = TRUE)
```



Based on the plot, it appears that the counties, represented by the data points, are grouped and colored based on quartiles. The x-axis in the first plot represents the first principal component (PC1), which explains a proportion of the variance in the data, and the y-axis represents the second principal component (PC2). In the second plot, the counties are projected onto the plane formed by PC2 and PC3.

The ellipses in the plot represent confidence regions that contain 95% of the data in each group, indicated by a centroid. However, it's observed that the ellipses overlap, which could be due to the artificial grouping of the continuous variable (Event) into quartiles.

The centroids of the ellipses show some slight separation along the x-axis for PC1 and marginally along the y-axis for PC2, indicating some degree of separation of groups based on the outcome of interest. However, in the second plot, the ellipses completely overlap, suggesting that removing unrelated variables from the correlation matrix may improve the analysis.

It's important to note that the data has not been cleaned yet, and examining the subset of extreme values for PC1 and PC2, which contain the largest variation in the data, may help identify any potential outliers and further refine the analysis

Problem 3.d (11 points)

- Based on the models we examined in the labs, fit an appropriate model with the aim to provide the most accurate prognosis you can for patients.
- Discuss and justify your decisions with several experiments and evidences.

```r
library(glmnet)
library(pROC)
library(knitr)

set.seed(1)

train.idx <- createDataPartition(nki.dt$Event, p = 0.7)$Resample
nki.train <- nki.dt[train.idx, ]
y <- nki.train$Event
# Exclude the target variable and IDs
x <- model.matrix( ~ . - Event, data = nki.train)
set.seed(1)
fit.ridge <-
  cv.glmnet(x,
            y,
            family = 'binomial',
            type.measure = 'auc',
            alpha = 0)
set.seed(1)
fit.lasso <-
  cv.glmnet(x, y, family = 'binomial', type.measure = 'auc')
nki.test <- nki.dt[-train.idx, ] # Use remaining data for testing
y.test <- nki.test$Event
x.test <- model.matrix( ~ . - Event, data = nki.test)
# Test predictions: Model S
y.test.s <- predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.min)
# Test predictions: Model B
y.test.b <- predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.1se)
# Test predictions: Lasso
y.test.lasso <-
  predict(fit.lasso, newx = x.test, s = fit.lasso$lambda.min)
# Test predictions: Ridge
y.test.ridge <-
  predict(fit.ridge, newx = x.test, s = fit.ridge$lambda.min)

```

```r
# Create ROC objects
roc.s <- roc(y.test, y.test.s)
```

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

Warning in roc.default(y.test, y.test.s): Deprecated use a matrix as predictor.
Unexpected results may be produced, please pass a numeric vector.

Setting direction: controls < cases
```

```r
roc.b <- roc(y.test, y.test.b)
```

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

Warning in roc.default(y.test, y.test.b): Deprecated use a matrix as predictor.
Unexpected results may be produced, please pass a numeric vector.

Setting direction: controls < cases
```

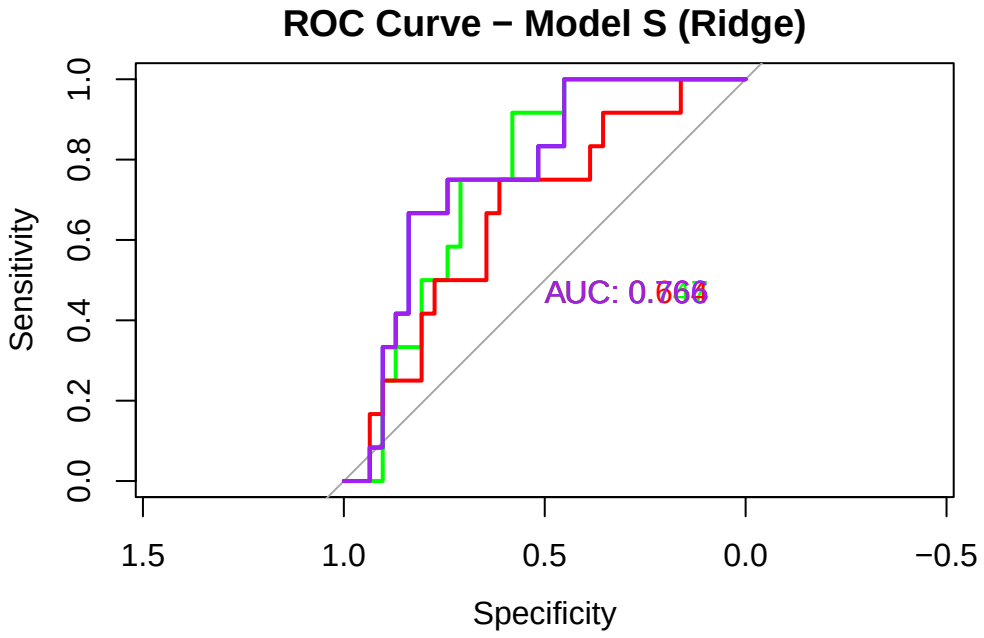
```r
roc.lasso <- roc(y.test, y.test.lasso)
```

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

Warning in roc.default(y.test, y.test.lasso): Deprecated use a matrix as
predictor. Unexpected results may be produced, please pass a numeric vector.

Setting direction: controls < cases
```

```r
roc.ridge <- roc(y.test, y.test.ridge)
```

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

Warning in roc.default(y.test, y.test.ridge): Deprecated use a matrix as
predictor. Unexpected results may be produced, please pass a numeric vector.

Setting direction: controls < cases
```

```r
# Plot ROC curves
plot(roc.s,
     col = "blue",
     main = "ROC Curve - Model S (Ridge)",
     print.auc = TRUE)
plot(roc.b,
     col = "green",
     add = TRUE,
     print.auc = TRUE)
plot(roc.lasso,
     col = "red",
     add = TRUE,
     print.auc = TRUE)
plot(roc.ridge,
     col = "purple",
     add = TRUE,
     print.auc = TRUE)
```




```r
# Create table
roc_table <- data.frame(
  Model = c("Model S", "Model B", "Lasso", "Ridge"),
  AUC = c(roc.s$auc, roc.b$auc, roc.lasso$auc, roc.ridge$auc)
```

Table 9: AUC values for different models

Model	AUC
Model S	0.7661290
Model B	0.7473118
Lasso	0.6639785
Ridge	0.7661290

```
)

# Print table using kable
kable(
  roc_table,
  caption = "AUC values for different models",
  col.names = c("Model", "AUC"),
  align = c("l", "c")
) %>% kable_styling(full_width = FALSE)
```

1. Model S: AUC = 0.7661290 Model S refers to the model obtained using stepwise selection with both forward and backward steps. The AUC value of 0.7661290 indicates that this model has a relatively good discriminatory ability for classifying events in the **`nki.dt`** dataset.
2. Model B: AUC = 0.7473118 Model B refers to the model obtained using backward elimination, where variables are removed from the full model in a stepwise manner. The AUC value of 0.7473118 indicates that this model also has good discriminatory ability, although slightly lower than Model S.
3. Lasso: AUC = 0.6639785 Lasso refers to the model obtained using the Lasso regularization method. The AUC value of 0.6639785 indicates that this model has relatively lower discriminatory ability compared to the stepwise selection models (Model S and Model B).
4. Ridge: AUC = 0.7661290 Ridge refers to the model obtained using the Ridge regularization method. The AUC value of 0.7661290 is the same as Model S, indicating that the Ridge model also has similar discriminatory ability to Model S.

In summary, Model S and Ridge models have the highest AUC values (0.7661290), indicating better discriminatory ability, followed by Model B (0.7473118), and Lasso (0.6639785) has the lowest AUC value among the models evaluated on the **`nki.dt`** dataset.