Assignment 4

Izzy Southon, 300597453

```r
# Read in dataset
eh <- read.csv("ATUSEH.csv")

# Restrict our focus to a subset of variables
eh <- eh[, c("EEINCOME1", "ERBMI", "ERTPREAT", "ERTSEAT", "EUDIETSODA",
"EUEXERCISE", "EUEXFREQ", "EUFASTFD", "EUFASTFDFRQ", "EUSNAP")]

head(eh)
```

```
##   EEINCOME1 ERBMI ERTPREAT ERTSEAT EUDIETSODA EUEXERCISE EUEXFREQ EUFASTFD
## 1         1  31.4       50       0         -1          1        5        2
## 2         2  25.7      120       0         -1          1        4        1
## 3         1  29.6       50      -2         -1          2       -1        1
## 4         3  23.4       95      30          2          1        6        1
## 5         1  35.9      140       5          1          2       -1        1
## 6         1  32.1       45       0         -1          1       10        1
##   EUFASTFDFRQ EUSNAP
## 1          -1      2
## 2           2      1
## 3           2      2
## 4           1      2
## 5           5      2
## 6           1      2
```

Assignment Questions

Question 1

a.

```r
# Overwrite EUEXFREQ
eh$EUEXFREQ <-ifelse(eh$EUEXERCISE == 2, 0, eh$EUEXFREQ)

# Overwrite EUFASTFDFRQ
eh$EUFASTFDFRQ <- ifelse(eh$EUFASTFD == 2, 0, eh$EUFASTFDFRQ)
```

b.

```r
# Overwrite EUDIETSODA
eh$EUDIETSODA <-ifelse(eh$EUDIETSODA == -1, 0, eh$EUDIETSODA)
```

c.

```r
# Convert missing data code values to NA
eh[eh == -1 | eh == -2 | eh == -3] <- NA

# Create table
missing_data_analysis <- data.frame(
  Frequency = sapply(eh, function(x) sum(is.na(x))),
  Proportion = sapply(eh, function(x) mean(is.na(x)))
)

missing_data_analysis$Proportion <- round(missing_data_analysis$Proportion, 5)

table(eh$EUEXFREQ)
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 3878  462 1086 1553 1018  926  346 1017   33   27   59    8   19    2   69    4
##   17   18   19   20   21   22   28   30   32   35   42   50
##    2    2    1    3    9    1    1    1    1    6    1    2
```

```r
sum(is.na(eh$EUEXFREQ))
```

```
## [1] 89
```

```r
print(missing_data_analysis)
```

```
##              Frequency Proportion
## EEINCOME1          239    0.02249
## ERBMI              555    0.05223
## ERTPREAT             0    0.00000
## ERTSEAT             61    0.00574
## EUDIETSODA           7    0.00066
## EUEXERCISE          69    0.00649
## EUEXFREQ            89    0.00838
## EUFASTFD            45    0.00423
## EUFASTFDFRQ         84    0.00791
## EUSNAP              84    0.00791
```

The variable that has the highest amount of missing data is ERBMI, where 0.05223 is the proportion of missing observations for ERBMI.

d.

```r
eh.complete <- na.omit(eh)

# eh has 10626 observations
# eh.complete has 9739 observations

# To calculate the proportion we take the complete
# dataset observations / the original datasets observations
round(1-(9739/10626),5)
```

```
## [1] 0.08347
```

In total, 0.08347 of the observations have been removed from the original dataset to produce this final dataframe.

e.

```r
eh.complete$OBESITY <- ifelse(eh.complete$ERBMI >= 30, 1, 0)

obesity_table <- table(eh.complete$OBESITY)

print(obesity_table)
```

```
##
##    0    1
## 6773 2966
```

Question 2

a.

```r
library(pander)
library(car)
```

```
## Loading required package: carData
```

```r
logistic.reg.model<-glm(OBESITY ~ EUEXFREQ + EUFASTFDFRQ + factor(EUDIETSODA)
                        + factor(EEINCOME1) + ERTPREAT + ERTSEAT
                        + factor(EUSNAP), family = "binomial", data = eh.complete)

# Calculate generalised inflation factors for predictors.
pander(vif(logistic.reg.model)) # Table 1
```

	GVIF	Df	GVIF^(1/(2*Df))
EUEXFREQ	1.018	1	1.009
EUFASTFDFRQ	1.045	1	1.022
factor(EUDIETSODA)	1.038	3	1.006
factor(EEINCOME1)	1.266	2	1.061
ERTPREAT	1.027	1	1.013
ERTSEAT	1.015	1	1.008
factor(EUSNAP)	1.23	1	1.109

To assess collinearity of predictors, we use variance inflation factors. Generalised VIF for the predictors that are not factors are all well under 10, and GVIF^(1/(2*Df)) are below 10 for predictors that are factors, alleviating any concerns related to multicollinearity.

b.

```r
# Table 2
pander(summary(logistic.reg.model))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.4317	0.08955	-4.821	1.429e-06
EUEXFREQ	-0.0884	0.00872	-10.14	3.784e-24
EUFASTFDFRQ	0.06585	0.009955	6.615	3.716e-11
factor(EUDIETSODA)1	0.6023	0.0761	7.915	2.482e-15
factor(EUDIETSODA)2	0.09559	0.06324	1.512	0.1307
factor(EUDIETSODA)3	1.005	0.2473	4.065	4.795e-05
factor(EEINCOME1)2	0.3809	0.05427	7.018	2.247e-12
factor(EEINCOME1)3	0.1431	0.107	1.338	0.181
ERTPREAT	-0.002661	0.0004905	-5.425	5.806e-08
ERTSEAT	-0.0009286	0.0004519	-2.055	0.03991
factor(EUSNAP)2	-0.3348	0.07541	-4.439	9.026e-06

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	11973 on 9738 degrees of freedom
Residual deviance:	11558 on 9728 degrees of freedom

Let,

- X_1 denote `EUEXFREQ`
- X_2 denote `EUFASTFDFRQ`
- X_3 denote `EUDIETSODA`
- X_4 denote `EEINCOME1`
- X_5 denote `ERTPREAT`
- X_6 denote `ERTSEAT`
- X_7 denote `EUSNAP`

Then the estimated logistic regression equation is,

$log(\frac{\hat{p}}{1-\hat{p}}) \approx -0.4317 - 0.0884X1 + 0.0659X2 + 0.6023X31 + 0.0956X32 + 1.0050X33 + 0.3809X42 + 0.1431X43 - 0.0027X5 - 0.0009X6 - 0.3348X72$

c.

Wald tests are shown in Table 2, which gives summary output for the model.

`EUEXFREQ`:

A test of

$H_0 : \beta_1 = 0$

$H_0 : \beta_1 \neq 0$

produces a test statistic of

$z = \frac{\widehat{\beta}_1}{SE(\widehat{\beta}_1)} \approx \frac{-0.0884}{0.0087} \approx -10.1400$

and a corresponding p-value of

$p - value = 2 \times P(Z > |-10.1400|) \approx 3.784 \times 10^{-24}$.

As the p-value is lower than any reasonable level of significance $\alpha = 0.05, 0.01$, we have enough evidence to suggest that β_1 is significantly different from 0 and there is a statistically significant relationship between the frequency of exercise in the past 7 days and obesity, adjusting for the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), the type of soft drink (if any) (`EUDIETSODA`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`). In particular, since the estimate for β_1 is negative, it indicates that the probability (as well as odds) of obesity decreases with increased frequency of exercise in the past 7 days after adjusting for `EUFASTFDFRQ`, `EUDIETSODA`, `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, and `EUSNAP`.

d.

Firstly, the exponentiate of the estimated coefficient needs to be obtained for `EUEXFREQ`,

$\widehat{\beta}_1 \approx -0.0884 \rightarrow exp(\widehat{\beta}_1) \approx 0.915$

```
# Table 3
pander(exp(confint.default(logistic.reg.model, parm = "EUEXFREQ")))
```

	2.5 %	97.5 %
EUEXFREQ	0.8999	0.9312

- An increase in the frequency of exercise in the past 7 days by one unit is associated with an estimated multiplicative change of 0.915 (95% CI: (0.8999, 0.9312)) in the odds of obesity, adjusting for the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), the type of soft drink (if any) (`EUDIETSODA`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`).

Alternatively, we can interpret this as percentage decrease in the odds of obesity. In particular, an increase in the frequency of exercise in the past 7 days by one unit is associated with a $(exp(-0.0884)-1)×100=-8.8 → $8.8%. (95% CI: (-10.01%, -6.88%)) decrease in the odds of obesity, adjusting for the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), the type of soft drink (if any) (`EUDIETSODA`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`).

e.

```
# Full model
full.model<-glm(OBESITY ~ EUEXFREQ
                + EUFASTFDFRQ
                + factor(EEINCOME1)
                + ERTPREAT
                + ERTSEAT
                + factor(EUSNAP)
```

```
                 + factor(EUDIETSODA),
                 family = "binomial", data = eh.complete)

# Fit the logistic regression model that excludes factor(EUDIETSODA).
reduced.model <- glm(OBESITY ~ EUEXFREQ
                      + EUFASTFDFRQ
                      + factor(EEINCOME1)
                      + ERTPREAT
                      + ERTSEAT
                      + factor(EUSNAP), family = "binomial", data = eh.complete)
```

If we let,

- X_1 denote `EUEXFREQ`
- X_2 denote `EUFASTFDFRQ`
- X_3 denote `EEINCOME1`
- X_4 denote `ERTPREAT`
- X_5 denote `ERTSEAT`
- X_6 denote `EUSNAP`
- X_7 denote `EUDIETSODA`

The full model is given by,

$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_{32} + \beta_4 X_{33} + \beta_5 X_4 + \beta_6 X_5 + \beta_7 X_{62} + \beta_8 X_{71} + \beta_9 X_{72} + \beta_{10} X_{73}$

and the reduced model is given by,

$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_{32} + \beta_4 X_{33} + \beta_5 X_4 + \beta_6 X_5 + \beta_7 X_{62}$

Thus a model comparison for these two models is the same as testing,

$H_0 : \beta_9 = \beta_{10} = 0$

$H_1 : \beta_9 \neq 0$ or $\beta_{10} \neq 0$,

```
# Carry out a likelihood ratio test comparing the full model to
# the model excluding EUDIETSODA as a predictor.

# Table 4
pander(anova(reduced.model, full.model, test = "LRT"), caption = "")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
9731	11633	NA	NA	NA
9728	11558	3	74.89	3.819e-16

The likelihood ratio test comparing these models is as shown in Table 4. The likelihood ratio test statistic is given by,

$G^2 \approx 74.89$,

which follows an asymptotic X_3^2 distribution under H_0. The p-value of p-value $\approx P(X_3^2 > 74.89) \approx 3.819 \times 10^{-16}$

is much smaller than any reasonable significance level ($\alpha = 0.05, 0.01$) meaning that we have sufficient evidence to conclude that either β_9 or β_{10} are significantly different from 0. This means that the inclusion of `EUDIETSODA` as a predictor leads to a significantly better fit after accounting for the other predictors in the model. This tells us that the types of soft drinks that a person consumes is important in predicting whether or not an individual is obese, and should be kept in the model.

f.

Based on the output from part (b),

Tests of

$H_0 : \beta_3 = 0$

$H_1 : \beta_3 \neq 0.$

and

$H_0 : \beta_4 = 0$

$H_1 : \beta_4 \neq 0.$

and

$H_0 : \beta_5 = 0$

$H_1 : \beta_5 \neq 0.$

produce test statistics of

$z \approx \frac{0.6023}{0.0761} \approx 7.915$

$z \approx \frac{0.0956}{0.0632} \approx 1.512$

$z \approx \frac{1.0050}{0.2473} \approx 4.065$

and corresponding p-values of

$p - value = 2 \times P(Z > |7.915|) \approx 2.482 \times 10^{-15}$

$p - value = 2 \times P(Z > |1.512|) \approx 0.1307$

$p - value = 2 \times P(Z > |4.065|) \approx 4.795 \times 10^{5}$

respectively. All p-values are less than $\alpha = 0.05$, meaning we have sufficient evidence to suggest that β_3, β_4, and β_5 are significantly different from 0. This means that we have sufficient evidence to suggest that there is a significant difference in the probability (as well as odds) of obesity for those who consumed soft drink that was diet and those who did not consume soft drink after adjusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food `(EUFASTFDFRQ)`, income `(EEINCOME1)`, time spent primarily eating and drinking `(ERTPREAT)`, time spent secondarily eating and drinking `(ERTSEAT)`, and food stamp benefits `(EUSNAP)` (this is indicated by the statistically significant result for the Wald test for β_3.

We also have sufficient evidence to suggest that there is a significant difference in the probability (as well as odds) of obesity for those who consumed soft drink that was regular and those who did not consume soft drink after djusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food `(EUFASTFDFRQ)`, income `(EEINCOME1)`, time spent primarily eating and drinking `(ERTPREAT)`, time spent secondarily eating and drinking `(ERTSEAT)`, and food stamp benefits `(EUSNAP)` (this is indicated by the statistically significant result for the Wald test for β_4.

We also have sufficient evidence to suggest that there is a significant difference in the probability (as well as odds) of obesity for those who consumed both kinds of soft drinks (diet and regular) and those who did not consume soft drink after djusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`) (this is indicated by the statistically significant result for the Wald test for β_5.

To interpret the effects corresponding to the coefficients for `EUDIETSODA`, we have to exponentiate the estimated coefficients.

$\hat{\beta}_3 \approx 0.6023 \rightarrow exp(\hat{\beta}_3) \approx 1.8263$

$\hat{\beta}_4 \approx 0.0956 \rightarrow exp(\hat{\beta}_4) \approx 1.1003$

$\hat{\beta}_5 \approx 1.0050 \rightarrow exp(\hat{\beta}_5) \approx 2.7319$

```
# Table 5
pander(exp(confint.default(logistic.reg.model, parm = c("factor(EUDIETSODA)1",
                                                         "factor(EUDIETSODA)2",
                                                         "factor(EUDIETSODA)3"))))
```

	2.5 %	97.5 %
factor(EUDIETSODA)1	1.573	2.12
factor(EUDIETSODA)2	0.972	1.246
factor(EUDIETSODA)3	1.683	4.436

The odds of obesity for those who consumed diet soft drink is estimated to be 1.826 (95% CI: (1.573, 2.120)) that of those who did not consume any soft drink, adjusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`).

The odds of obesity for those who consumed regular soft drink is estimated to be 1.1003 (95% CI: (0.972, 1.246)) that of those who did not consume any soft drink, adjusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`).

The odds of obesity for those who consumed both types of soft drink is estimated to be 2.7319 (95% CI: (1.683, 4.436)) that of those who did not consume any soft drink, adjusting for the number of times in the past 7 days someone participated in physical activities (`EUEXFREQ`), the number of times in the past 7 days food was purchased from a deli, carry-out, delivery food, or fast food (`EUFASTFDFRQ`), income (`EEINCOME1`), time spent primarily eating and drinking (`ERTPREAT`), time spent secondarily eating and drinking (`ERTSEAT`), and food stamp benefits (`EUSNAP`).

g.

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-6   2023-06-27
```

```r
pander(hoslem.test(eh.complete$OBESITY, logistic.reg.model$fitted.values, g = 10))
```

Table 7: Hosmer and Lemeshow goodness of fit (GOF) test: `eh.complete$OBESITY, logistic.reg.model$fitted.values`

Test statistic	df	P value
6.685	8	0.571

```r
pander(hoslem.test(eh.complete$OBESITY, logistic.reg.model$fitted.values, g = 20))
```

Table 8: Hosmer and Lemeshow goodness of fit (GOF) test: `eh.complete$OBESITY, logistic.reg.model$fitted.values`

Test statistic	df	P value
18.71	18	0.4101

```r
pander(hoslem.test(eh.complete$OBESITY, logistic.reg.model$fitted.values, g = 30))
```

Table 9: Hosmer and Lemeshow goodness of fit (GOF) test: `eh.complete$OBESITY, logistic.reg.model$fitted.values`

Test statistic	df	P value
38.76	28	0.08482

When g = 10, 20 or 30, the p-values fluctuate between 0.0848 and 0.5710. All p-values are greater than $\alpha = 0.05, 0.01$, indicating that the model provides a reasonable fit to the obesity data.

Question 3

EDA including the variables `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, `EUEXERCISE`, `EUEXFREQ`, `EUFASTFD`, `EUFASTFDFRQ`, and `EUSNAP`.

a.

```r
library(MASS)

# Forward selection
forward.selection.obesity <- stepAIC(glm(OBESITY ~ 1, family = "binomial", data =
eh.complete), scope = list(upper = ~ factor(EEINCOME1) + ERTPREAT + ERTSEAT
                           + factor(EUDIETSODA) + factor(EUEXERCISE) + EUEXFREQ
                           + factor(EUFASTFD) + EUFASTFDFRQ + factor(EUSNAP), lower = ~1),
direction = "forward", trace = FALSE)

pander(forward.selection.obesity$anova)
```

Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	NA	NA	9738	11973	11975
+ EUEXFREQ	1	160.6	9737	11812	11816
+ factor(EEINCOME1)	2	75.25	9735	11737	11745
+ factor(EUDIETSODA)	3	82.44	9732	11654	11668
+ EUFASTFDFRQ	1	40.9	9731	11613	11629
+ ERTPREAT	1	31.42	9730	11582	11600
+ factor(EUSNAP)	1	19.5	9729	11562	11582
+ factor(EUFASTFD)	1	9.732	9728	11553	11575
+ factor(EUEXERCISE)	1	7.583	9727	11545	11569
+ ERTSEAT	1	4.302	9726	11541	11567

```r
# Backward selection
backward.selection.obesity <- stepAIC(glm(OBESITY ~ factor(EEINCOME1)
                                          + ERTPREAT
                                          + ERTSEAT
                                          + factor(EUDIETSODA)
                                          + factor(EUEXERCISE)
                                          + EUEXFREQ
                                          + factor(EUFASTFD)
                                          + EUFASTFDFRQ
                                          + factor(EUSNAP),
                                          family = "binomial",
                                          data = eh.complete),
                                      scope = list(upper = ~ factor(EEINCOME1)
                                                   + ERTPREAT
                                                   + ERTSEAT
                                                   + factor(EUDIETSODA)
                                                   + factor(EUEXERCISE)
                                                   + EUEXFREQ
                                                   + factor(EUFASTFD)
                                                   + EUFASTFDFRQ
                                                   + factor(EUSNAP),
                                                   lower = ~1),
                                      direction = "backward", trace = FALSE)

pander(backward.selection.obesity$anova)
```

Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	NA	NA	9726	11541	11567

Both forward and backward selection arrive at a model that says all predictors included in the model are important in predicting obesity. This is because the AIC rules of thumb say that model B (the model with more predictors) is preferred. Forward and backward selection may not arrive at the same optimal model because forward selection starts from the null model, and backward selection starts with the full model. Forward selection adds important predictors to the null model, whereas backward selection removes less relevant predictors, therefore meaning they may not arrive at the same optimal model. Even if forward and backward selection arrive at the same model, it is not guaranteed that it is in fact the optimal model.

b.

```r
library(bestglm)
```

```
## Loading required package: leaps
```

```r
# Construct a dataframe

predictors.for.bestglm <- data.frame(EEINCOME1 = as.factor(eh.complete$EEINCOME1),
                                     ERTPREAT = eh.complete$ERTPREAT,
                                     ERTSEAT = eh.complete$ERTSEAT,
                                     EUDIETSODA = as.factor(eh.complete$EUDIETSODA),
                                     EUEXERCISE = as.factor(eh.complete$EUEXERCISE),
                                     EUEXFREQ = (eh.complete$EUEXFREQ),
                                     EUFASTFD = as.factor(eh.complete$EUFASTFD),
                                     EUFASTFDFRQ = eh.complete$EUFASTFDFRQ,
                                     EUSNAP = as.factor(eh.complete$EUSNAP),
                                     y = eh.complete$OBESITY)

# Find the best logistic regression model based on the predictors according
# to the criterion of minimising AIC.
best.logistic.AIC <- bestglm(Xy = predictors.for.bestglm,
                             family = binomial, IC = "AIC",
                             method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
```

```
## Note: factors present with more than 2 levels.
```

```r
# Show the top five models in terms of minimising AIC.
pander(best.logistic.AIC$BestModels)
```

Table 12: Table continues below

EEINCOME1	ERTPREAT	ERTSEAT	EUDIETSODA	EUEXERCISE	EUEXFREQ	EUFASTFD
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE

EUFASTFDFRQ	EUSNAP	Criterion
TRUE	TRUE	11565
TRUE	TRUE	11567
TRUE	TRUE	11570
TRUE	TRUE	11573
TRUE	TRUE	11574

```r
# Find the best logistic regression model based on the predictors according
# to the criterion of minimising BIC.
best.logistic.BIC <- bestglm(Xy = predictors.for.bestglm,
                             family = binomial, IC = "BIC",
                             method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
# Show the top five models in terms of minimising BIC.
pander(best.logistic.BIC$BestModels)
```

Table 14: Table continues below

EEINCOME1	ERTPREAT	ERTSEAT	EUDIETSODA	EUEXERCISE	EUEXFREQ	EUFASTFD
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE

EUFASTFDFRQ	EUSNAP	Criterion
TRUE	TRUE	11645
TRUE	TRUE	11645
TRUE	TRUE	11646
TRUE	TRUE	11648
FALSE	TRUE	11648

In terms of minimising AIC, the AIC criteria finds the optimal model includes all predictors `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, `EUEXERCISE`, `EUEXFREQ`, `EUFASTFD`, `EUFASTFDFRQ`, and `EUSNAP`.

In terms of minimising BIC, the BIC criteria finds the optimal model includes the predictors `EEINCOME1`, `ERTPREAT`, `EUDIETSODA`, `EUEXFREQ`, `EUFASTFD`, `EUFASTFDFRQ`, and `EUSNAP`. BIC drops the predictors `ERTSEAT` and `EUEXERCISE` from the optimal model.

The optimal models obtained from AIC and BIC are different as the AIC criteria includes the predictors `ERTSEAT` and `EUEXERCISE`, but the BIC criteria excludes them from its optimal model. The reason why the criteria of AIC and BIC may lead to different "best" model is because the penalty term used by BIC depends on the sample size (penalty is $log(n)$ for BIC, where compared to AIC it is 2) and is larger or larger sample sizes. This means that a bigger reduction in deviance is required to warrant the "cost" of an additional parameter.

Using AIC, the best model matches that of the forward and backward selection algorithm which the optimal model was the one that included all predictors `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, `EUEXERCISE`, `EUEXFREQ`, `EUFASTFD`, `EUFASTFDFRQ`, and `EUSNAP`. Using BIC, the best model did not match that of the forward and backward selection algorithm. This is because best subset selection considers all possible subsets then selects the one with the best predictive performance (based on AIC or BIC). As mentioned, BIC has a higher penalty and is likely to select fewer predictors like it has in this example using the obesity dataset.

c.

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(doParallel)
```

```
## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```r
library(foreach)

# Convert factors into factors
eh.complete$EEINCOME1 <- factor(eh.complete$EEINCOME1)
eh.complete$EUDIETSODA <- factor(eh.complete$EUDIETSODA)
eh.complete$EUEXERCISE <- factor(eh.complete$EUEXERCISE)
eh.complete$EUFASTFD <- factor(eh.complete$EUFASTFD)
eh.complete$EUSNAP <- factor(eh.complete$EUSNAP)

head(eh.complete) # Checking the indices of predictors
```

```
##   EEINCOME1 ERBMI ERTPREAT ERTSEAT EUDIETSODA EUEXERCISE EUEXFREQ EUFASTFD
## 1         1  31.4       50       0          0          1        5        2
## 2         2  25.7      120       0          0          1        4        1
## 4         3  23.4       95      30          2          1        6        1
## 5         1  35.9      140       5          1          2        0        1
## 6         1  32.1       45       0          0          1       10        1
## 7         1  30.1      105       2          1          2        0        2
##   EUFASTFDFRQ EUSNAP OBESITY
## 1           0      2       1
## 2           2      1       0
## 4           1      2       0
## 5           5      2       1
## 6           1      2       1
## 7           0      2       1
```

```r
# Specify the indices of the variables to be considered in predictive models for presence of obesity in
variable.indices <- c(1,3,4,5,6,7,8,9,10) # indice 2 is ERBMI which is excluded, 11 is OBESITY the resp

# Produce a matrix that represents all possible combinations of variables
all.comb <- expand.grid(as.data.frame(matrix(rep(0 : 1, length(variable.indices)), nrow = 2)))[-1, ]

head(all.comb) # 2^9 - 1 = 511 possible models
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9
## 2  1  0  0  0  0  0  0  0  0
```

```
## 3  0  1  0  0  0  0  0  0  0
## 4  1  1  0  0  0  0  0  0  0
## 5  0  0  1  0  0  0  0  0  0
## 6  1  0  1  0  0  0  0  0  0
## 7  0  1  1  0  0  0  0  0  0
```

```
nrow(all.comb)
```

```
## [1] 511
```

```
# Specify number of folds and reps
folds <- 10
reps <- 20

nclust <- makeCluster(detectCores() * 0.75)
registerDoParallel(nclust)

# Accuracy
fitControl <- trainControl(method = "repeatedcv", number = folds,
repeats = reps, seeds = 1 : (folds * reps + 1), classProbs = TRUE,
savePredictions = TRUE)

# Save estimated accuracy and standard deviation for each model type and set of covariates
accuracy <- foreach(i = 1 : nrow(all.comb), .combine = "rbind",
.packages = "caret") %dopar%
{
c(i, unlist(train(as.formula(paste("make.names(OBESITY) ~",
paste(names(eh.complete)[variable.indices][all.comb[i,] == 1],
collapse = " + "))), data = eh.complete, trControl = fitControl,
method = "glm", family = "binomial")$results[c(2, 4)]))
}

rownames(accuracy) <- NULL

# Area under the roc curve
fitControl <- trainControl(method = "repeatedcv", number = folds,
repeats = reps, seeds = 1 : (folds * reps + 1), summaryFunction =
twoClassSummary, classProbs = TRUE, savePredictions = TRUE)

# Save estimated AUC and standard deviation for each model type and set of covariates, using untransfor
AUC <- foreach(i = 1 : nrow(all.comb), .combine = "rbind", .packages =
"caret") %dopar%
{
c(i, unlist(train(as.formula(paste("make.names(OBESITY) ~",
paste(names(eh.complete)[variable.indices][all.comb[i,] == 1],
collapse = " + "))), data = eh.complete, trControl = fitControl,
method = "glm", family = "binomial", metric = "ROC")$results[c(2, 5)]))
}

rownames(AUC) <- NULL

# i. view the model that maximises accuracy (minimising total error rate)
names(eh.complete)[variable.indices[all.comb[which.max(accuracy[,2]),] == 1]]
```

```
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE"
## [6] "EUFASTFD"   "EUSNAP"
```

```r
# ii. view the model that maximises AUC
names(eh.complete)[variable.indices[all.comb[which.max(AUC[, 2]),] == 1]]
```

```
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFDFRQ" "EUSNAP"
```

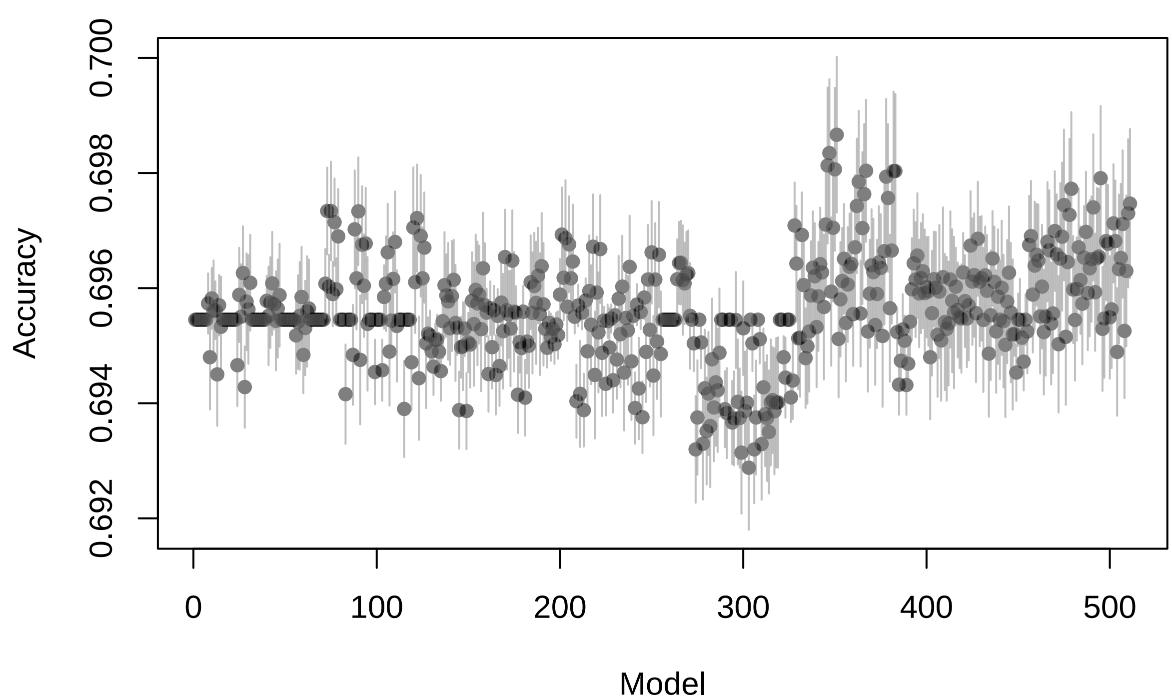
```r
### Accuracy

# View all models within one SE of the best model.
best.models.accuracy <- (1 : nrow(all.comb))[accuracy[, 2] + accuracy[, 3] / sqrt(reps) >=
max(accuracy[, 2])]

for(i in 1 : length(best.models.accuracy))
{
cat(paste("Model ", i, ":\n"))
print(names(eh.complete)[variable.indices[all.comb[best.models.accuracy[i], ] == 1]]) #
print(accuracy[best.models.accuracy[i], 2]) # Accuracy

cat("\n")
}
```

```
## Model  1 :
## [1] "ERTPREAT"   "EUDIETSODA" "EUEXERCISE" "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6981315
##
## Model  2 :
## [1] "EEINCOME1"  "ERTPREAT"   "EUDIETSODA" "EUEXERCISE" "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6983472
##
## Model  3 :
## [1] "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE" "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6980648
##
## Model  4 :
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6986656
##
## Model  5 :
## [1] "EEINCOME1"  "ERTPREAT"   "EUDIETSODA" "EUEXFREQ"   "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6978493
##
```

```
## Model  6 :
## [1] "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXFREQ"   "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6976335
##
## Model  7 :
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXFREQ"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6980392
##
## Model  8 :
## [1] "ERTPREAT"   "EUDIETSODA" "EUEXERCISE" "EUEXFREQ"   "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6979363
##
## Model  9 :
## [1] "EEINCOME1"  "ERTPREAT"   "EUDIETSODA" "EUEXERCISE" "EUEXFREQ"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6975668
##
## Model  10 :
## [1] "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE" "EUEXFREQ"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6980339
##
## Model  11 :
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE"
## [6] "EUEXFREQ"   "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6980341
##
## Model  12 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXERCISE"  "EUFASTFD"
## [6] "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6974437
##
## Model  13 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6977261
##
## Model  14 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXFREQ"    "EUFASTFD"
## [6] "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6974026
##
```

```
## Model  15 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6979109
##
## Model  16 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUEXFREQ"    "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6974694
```

```
# Plot point estimates for accuracy for each model.
plot(accuracy[, 2], xlab = "Model", ylim = c(min(accuracy[, 2] - accuracy[, 3] / sqrt(reps)),
max(accuracy[, 2] + accuracy[, 3] / sqrt(reps))), ylab = "Accuracy", pch = 16, col =gray(0, alpha =
0.5))

# Include vertical lines extending one standard error above and below the accuracy for each model.
for(i in 1 : nrow(all.comb))
{
points(rep(i, 2), c(accuracy[i, 2] - accuracy[i, 3] / sqrt(reps), accuracy[i, 2] + accuracy[i, 3] /
sqrt(reps)), type = "l", col = gray(0.5, alpha = 0.5))
}
```



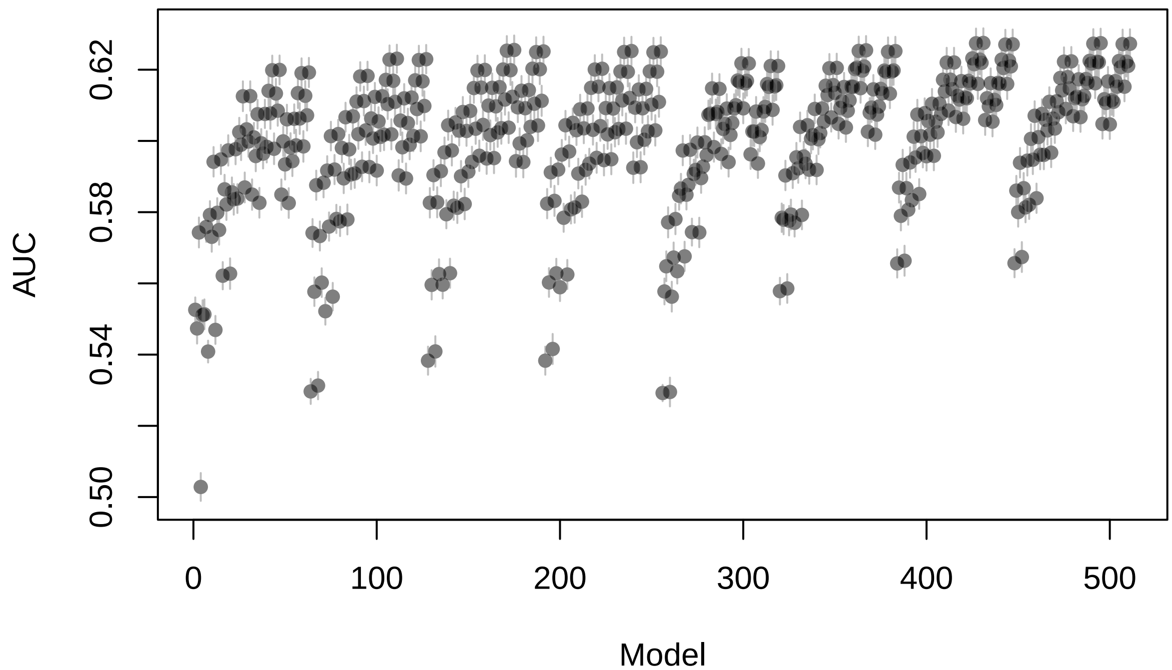
```r
### AUC

# View all models within one SE of the best model.
best.models.AUC <- (1 : nrow(all.comb))[AUC[, 2] + AUC[, 3] / sqrt(reps) >= max(AUC[, 2])]

for(i in 1 : length(best.models.AUC))
{
cat(paste("Model ", i, ":\n"))
print(names(eh.complete)[variable.indices[all.comb[best.models.AUC[i], ] == 1]])
print(accuracy[best.models.AUC[i], 2]) # AUC

cat("\n")
}
```

```
## Model  1 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXFREQ"    "EUFASTFDFRQ"
##  Accuracy
## 0.6956159
##
## Model  2 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFDFRQ"
##  Accuracy
## 0.6955902
##
## Model  3 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXERCISE"  "EUEXFREQ"
## [6] "EUFASTFDFRQ"
##  Accuracy
## 0.6957442
##
## Model  4 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUEXFREQ"    "EUFASTFDFRQ"
##  Accuracy
## 0.6957187
##
## Model  5 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXFREQ"    "EUFASTFD"
## [6] "EUFASTFDFRQ"
##  Accuracy
## 0.6945326
##
## Model  6 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFD"    "EUFASTFDFRQ"
##  Accuracy
## 0.6947277
##
## Model  7 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXERCISE"  "EUEXFREQ"
## [6] "EUFASTFD"    "EUFASTFDFRQ"
##  Accuracy
```

```
## 0.6944812
##
## Model  8 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUEXFREQ"    "EUFASTFD"    "EUFASTFDFRQ"
##  Accuracy
## 0.6948559
##
## Model  9 :
## [1] "EEINCOME1"  "ERTPREAT"   "EUDIETSODA" "EUEXFREQ"   "EUFASTFD"
## [6] "EUSNAP"
##  Accuracy
## 0.6978493
##
## Model  10 :
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXFREQ"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6980392
##
## Model  11 :
## [1] "EEINCOME1"  "ERTPREAT"   "EUDIETSODA" "EUEXERCISE" "EUEXFREQ"
## [6] "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6975668
##
## Model  12 :
## [1] "EEINCOME1"  "ERTPREAT"   "ERTSEAT"    "EUDIETSODA" "EUEXERCISE"
## [6] "EUEXFREQ"   "EUFASTFD"   "EUSNAP"
##  Accuracy
## 0.6980341
##
## Model  13 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXFREQ"    "EUFASTFDFRQ"
## [6] "EUSNAP"
##  Accuracy
## 0.6955647
##
## Model  14 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6954466
##
## Model  15 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXERCISE"  "EUEXFREQ"
## [6] "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6950155
##
## Model  16 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUEXFREQ"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
```

```
## 0.6951953
##
## Model  17 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXFREQ"    "EUFASTFD"
## [6] "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6974026
##
## Model  18 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXFREQ"
## [6] "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6979109
##
## Model  19 :
## [1] "EEINCOME1"   "ERTPREAT"    "EUDIETSODA"  "EUEXERCISE"  "EUEXFREQ"
## [6] "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6971151
##
## Model  20 :
## [1] "EEINCOME1"   "ERTPREAT"    "ERTSEAT"     "EUDIETSODA"  "EUEXERCISE"
## [6] "EUEXFREQ"    "EUFASTFD"    "EUFASTFDFRQ" "EUSNAP"
##  Accuracy
## 0.6974694
```

```r
# Plot point estimates for AUC for each model.
plot(AUC[, 2], xlab = "Model", ylim = c(min(AUC[, 2] - AUC[, 3] / sqrt(reps)), max(AUC[, 2] + AUC[,
3] / sqrt(reps))), ylab = "AUC", pch = 16, col = gray(0, alpha = 0.5))
# Include vertical lines extending one standard error above and below the AUC for each model.
for(i in 1 : nrow(all.comb))
{
points(rep(i, 2), c(AUC[i, 2] - AUC[i, 3] / sqrt(reps), AUC[i, 2] + AUC[i, 3] / sqrt(reps)), type =
"l", col = gray(0.5, alpha = 0.5))
}
```



To minimize total error rate you want to maximize accuracy. The optimal model selected according to the criteria of minimising total error rate includes the predictors `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, `EUEXERCISE`, `EUFASTFD`, and `EUSNAP` (model 4).

The optimal model selected according to the criteria of maximising AUC includes the predictors `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, `EUEXFREQ`, `EUFASTFDFRQ`, and `EUSNAP` (model 14).

The optimal models depending on these two types of criteria are different. They are different as they both include the predictors `EEINCOME1`, `ERTPREAT`, `ERTSEAT`, `EUDIETSODA`, and `EUSNAP`. But the criteria of minimising total error rate selects `EUEXERCISE` and `EUFASTFD` as important predictors, whereas the criteria of maximising AUC selects `EUEXFREQ` and `EUFASTFDFRQ` as important predictors.

Total error rate is the proportion of all predictions that are incorrect, and excludes predictors that do not significantly contribute to lowering the overall error. AUC measures the area under the ROC curve, which plots the true positive rate against the false positive rate. Different optimal models might be obtained based on these different criteria because minimising total error rate needs a specific threshold to be chosen, which may not be the same that maximizes AUC. Total error rate is more sensitive to imbalanced datasets compared to AUC which is more robust to imbalanced datasets. Total error rate treats false positives and false negatives equally whereas AUC accounts for trade-offs between TPR and FPR.Predictors that do not contribute significantly to improving the AUC-ROC curve performance, such as EUEXERCISE and EUFASTFD, are excluded.This explains why the minimising total error rate and maximising AUC may lead to different "best" models.

In part (a), both forward and backward selection arrived at a model that said all predictors included in the model are important in predicting obesity. In part (b), the criteria of minimising AIC found the optimal model that included all predictors. The criteria of minimising BIC found the optimal model that dropped the predictors `ERTSEAT` and `EUEXERCISE` from the optimal model. In part (c), the critera of minimising total error dropped the predictors `EUEXFREQ` and `EUFASTFDFRQ`. The criteria of maximising AUC dropped the predictors `EUEXERCISE` and `EUFASTFD`. In summary, optimal models differ because cross-validation assesses predictive performance on unseen data, leading to different predictors being selected or dropped based on how well they minimize error or maximize AUC on multiple data splits. Practically, this evaluation reveals a more reliable set of predictors compared to forward and backward selection or AIC and BIC criteria, which may overfit the data (where the model does not perform well on new, unseen data). Because of this difference and part (c)'s ability to generalise to new, unseen data, it is not surprising that different optimal models have been found in part (c) compared to parts (a) and (b).