Assignment II

Group2

2024-12-12

Introduction

Question 1:

Create table 1. Describe all available variables in the data. Show both, the original data and the imputed data.

Overview:

All available variables in the data are as follows:

- *agegroup:* Categories for age.
- *smoker:* Binary variable for smoker or not.
- *smokerf:* Smoker factor with levels No and Yes.
- *heightcm:* Convert height from inches to cm.
- *weightkg:* Convert weight from pounds to kg.
- *bmi:* Calculate BMI.
- *bmicat:* Categories for BMI.
- *cholmmol:* Convert cholesterol from mg/dl to mmol/l.
- *sbp10:* Categories of sbp (systolic blood pressure).
- *sbpcat:* Systolic blood pressure factor.
- *dibpat0f:* Dichotomous behavior pattern factor with levels A and B instead of 1 and 0. A classification system where individuals are grouped into one of two distinct categories based on their behavioral traits
- *arcus0:* Corneal arcus factor which is caused by lipid deposits in the cornea. It's presence may indicate high cholesterol levels and increased risk of heart disease.
- *chd69f:* Coronary heart disease factor.

Then, we have created tables for the original data. In the next step, we have imputed the data using Multivariate Imputation and created tables for the imputed data.

Code:

```
# Define variables
variables <- c("id", "agegroup", "age0", "cholmmol", "sbp10", "bmi", "smokerf",
               "arcus0", "dibpat0f", "chd69")
categorical <- c("smokerf", "dibpat0f", "chd69")

# Create Table 1 for the original data
table_original <- CreateTableOne(vars = variables, data = wcgs, factorVars = categorical)
```

```r
# Create Table 1 for the imputed data
table_imputed <- CreateTableOne(vars = variables, data = di, factorVars = categorical)
```

Output:

```
##
##      ### Summary of continuous variables ###
##
## strata: Overall
##             n miss p.miss  mean    sd median  p25   p75  min   max skew kurt
## id       3154    0   0.00 1e+04 6e+03  11406 3741 13115 2001 22101  0.2 -0.7
## age0     3154    0   0.00 5e+01 6e+00     45   42    50   39    59  0.5 -0.8
## cholmmol 3154   13   0.41 6e+00 1e+00      6    5     6    3    11  0.4  0.5
## sbp10    3154    0   0.00 1e+01 2e+00     13   12    14   10    23  1.2  2.8
## bmi      3154    0   0.00 2e+01 3e+00     24   23    26   11    39  0.5  2.0
## arcus0   3154    2   0.06 3e-01 5e-01      0    0     1    0     1  0.9 -1.2
##
## ================================================================================
##
##      ### Summary of categorical variables ###
##
## strata: Overall
##       var    n miss p.miss   level freq percent cum.percent
##  agegroup 3154    0    0.0 [39,45) 1448    45.9        45.9
##                            [45,55) 1384    43.9        89.8
##                            [55,60]  322    10.2       100.0
##
##   smokerf 3154    0    0.0      No 1652    52.4        52.4
##                               Yes 1502    47.6       100.0
##
##  dibpat0f 3154    0    0.0       B 1565    49.6        49.6
##                                  A 1589    50.4       100.0
##
##     chd69 3154    0    0.0       0 2897    91.9        91.9
##                                  1  257     8.1       100.0
##

##
##      ### Summary of continuous variables ###
##
## strata: Overall
##             n miss p.miss  mean    sd median  p25   p75  min   max skew kurt
## id       3154    0      0 1e+04 6e+03  11406 3741 13115 2001 22101  0.2 -0.7
## age0     3154    0      0 5e+01 6e+00     45   42    50   39    59  0.5 -0.8
## cholmmol 3154    0      0 6e+00 1e+00      6    5     6    3    11  0.4  0.5
## sbp10    3154    0      0 1e+01 2e+00     13   12    14   10    23  1.2  2.8
## bmi      3154    0      0 2e+01 3e+00     24   23    26   11    39  0.5  2.0
## arcus0   3154    0      0 3e-01 5e-01      0    0     1    0     1  0.9 -1.2
##
## ================================================================================
##
##      ### Summary of categorical variables ###
##
## strata: Overall
```

```
##       var    n miss p.miss   level freq percent cum.percent
##  agegroup 3154    0    0.0 [39,45) 1448    45.9        45.9
##                            [45,55) 1384    43.9        89.8
##                            [55,60]  322    10.2       100.0
##
##   smokerf 3154    0    0.0      No 1652    52.4        52.4
##                               Yes 1502    47.6       100.0
##
##  dibpat0f 3154    0    0.0       B 1565    49.6        49.6
##                                  A 1589    50.4       100.0
##
##     chd69 3154    0    0.0       0 2897    91.9        91.9
##                                  1  257     8.1       100.0
##
```

Conclusion:

The imputed data has been created using Multivariate Imputation where the missing data of cholmmol has been imputed. The imputed data has been created using Predictive Mean Matching (PMM) method.

Question 2:

Calculate the overall risk of CHD in the cohort.

Overview:

a. What is the outcome we are interested in?

The outcome we are interested in is Coronary Heart Disease (CHD).

b. What are the known risk factors for our outcome of interest?

The known risk factors for Coronary Heart Disease (CHD) are as follows:

- Dichotomous Behaviour type A/B (dibpat0f)
- Age (agegroup, age0)
- Cholesterol (cholmmol)
- Systolic Blood Pressure (sbp10)
- BMI (bmi)
- Smoking (smokerf)
- Corneal arcus (arcus0)

c. How many persons are included?

3154 middle-aged men, from 39 to 59 years of age, during the years 1960-1961 are included in this prospective cohort study.

d. What is the overall risk or rate and prevalence of the disease in our cohort?

The overall risk or rate and prevalence of the disease in our cohort is as follows:

```r
# Overall risk or rate
overall_rate <- table(di$chd69)

#calculate risk of CHD
overall_risk <- overall_rate / sum(overall_rate)

# extract the rate and risk into a data frame
```

```r
chd_frame <- data.frame(
  "CHD Presence" = c("No", "Yes"),
  "Overall Rate" = c(as.matrix(overall_rate)[1], as.matrix(overall_rate)[2]),
  "Overall Risk" = c(as.matrix(overall_risk)[1], as.matrix(overall_risk)[2])
)
#print overall rate and risk into a table
knitr::kable(chd_frame, col.names = c("CHD Presence", "Overall Rate", "Overall Risk"))
```

CHD Presence	Overall Rate	Overall Risk
No	2897	0.9185162
Yes	257	0.0814838

Analysis:

The overall risk of Coronary Heart Disease (CHD) in the cohort is 0.08, which indicates that prevalence of the disease is 8% in the cohort.

Question 3:

Overview:

To solve this problem, we need to build an optimal prediction model for the outcome of Coronary Heart Disease (CHD) using the available data. We will use logistic regression due to the binary nature of the outcome and select predictors that improve our predictions. Additionally, we will consider interaction terms and ensure that categorical variables are appropriately handled.

3.a. Building the Optimal Prediction Model:

Step 1: Model Selection: Logistic regression is suitable for predicting Coronary Heart Disease (CHD) because:

- Binary outcome: CHD is a binary outcome, meaning it can be either present (1) or absent (0). Logistic regression is designed to model binary outcomes.
- Multiple predictors: There are multiple known risk factors for CHD, and logistic regression can handle multiple predictor variables.
- Quantification of risk: Logistic regression can provide estimates of the probability of developing CHD based on the values of the predictor variables, which can be useful for risk assessment and decision-making.

The `rms` package in R provides functions for regression modeling strategies, including logistic regression via the `lrm` function.

Step 2: Variable Selection: We start by fitting a full model that includes all potential predictors:

```r
dd <- datadist(di)
options(datadist="dd")
full_model <- lrm(chd69 ~ dibpat0f + age0 + cholmmol + sbp10 + bmi + smokerf
                  + arcus0, data=di, x=TRUE, y=TRUE)
# Extract the model summary
model_summary <- as.data.frame(summary(full_model))
knitr::kable(model_summary, col.names = c("Variable", "Low", "High", "Diff", "Effect",
                                          "S.E.", "Lower 95%", "Upper 95%"),
```

```
            align = c("l", "c", "c", "c", "c", "c", "c", "c"),
            caption = "Summary of the Logistic Regression Model")
```

Table 2: Summary of the Logistic Regression Model

	Variable	Low	High	Diff	Effect	S.E.	Lower 95%	Upper 95%
age0	42.000000	50.00000	8.000000	0.4444889	0.0972089	0.2539630	0.6350148	1
X.Odds.Ratio	42.000000	50.00000	8.000000	1.5596929	NA	1.2891241	1.8870501	2
cholmmol	5.057692	6.48718	1.429487	0.5778017	0.0853722	0.4104753	0.7451282	1
X.Odds.Ratio.1	5.057692	6.48718	1.429487	1.7821166	NA	1.5075341	2.1067115	2
sbp10	12.000000	13.60000	1.600000	0.2959951	0.0654344	0.1677460	0.4242442	1
X.Odds.Ratio.2	12.000000	13.60000	1.600000	1.3444636	NA	1.1826362	1.5284347	2
bmi	22.957374	25.84272	2.885343	0.1626619	0.0761162	0.0134768	0.3118469	1
X.Odds.Ratio.3	22.957374	25.84272	2.885343	1.1766388	NA	1.0135681	1.3659455	2
arcus0	0.000000	1.00000	1.000000	0.2437805	0.1422739	-0.0350713	0.5226323	1
X.Odds.Ratio.4	0.000000	1.00000	1.000000	1.2760642	NA	0.9655366	1.6864611	2
dibpat0f...B.A	2.000000	1.00000	NA	-0.7051579	0.1442429	-0.9878687	-0.4224471	1
X.Odds.Ratio.5	2.000000	1.00000	NA	0.4940305	NA	0.3723695	0.6554409	2
smokerf...Yes.Nd	.000000	2.00000	NA	0.5773367	0.1408234	0.3013278	0.8533455	1
X.Odds.Ratio.6	1.000000	2.00000	NA	1.7812879	NA	1.3516524	2.3474872	2

```
AIC(full_model)
```

```
## [1] 1609.815
```

3.b. Including Interaction Terms

We need to check if including interaction terms between certain predictors improves the model fit. When considering interaction terms in a logistic regression model, one needs to think which variables might have a combined effect on the outcome (Coronary Heart Disease) that's different from their individual effects. Here are some potential interaction terms along with their rationale:

- **Age and Cholesterol:** As people age, their cholesterol levels may have a greater impact on their risk of Coronary Heart Disease. This interaction term can help capture the potential synergistic effect of increasing age and cholesterol levels.
- **Smoking and Age:** Smoking is a well-known risk factor for Coronary Heart Disease, and its effects may be exacerbated with increasing age. This interaction term can help account for the potential increased risk of Coronary Heart Disease among older smokers.
- **BMI and Systolic Blood Pressure:** High blood pressure is often associated with obesity, and the combination of these two factors may increase the risk of Coronary Heart Disease more than either factor alone. This interaction term can help capture the potential additive effect of high BMI and systolic blood pressure.
- **Cholesterol and Systolic Blood Pressure:** High cholesterol and high blood pressure are both risk factors for Coronary Heart Disease, and their combined effect may be greater than the sum of their individual effects. This interaction term can help account for the potential synergistic effect of these two factors.
- **Corneal arcus and Age:** Corneal arcus is a sign of lipid deposition in the cornea, which may be associated with increased risk of Coronary Heart Disease. The effect of corneal arcus may be more pronounced in older individuals, making this interaction term a potential candidate.
- **Smoking and Cholesterol:** Smoking can increase cholesterol levels, and the combination of these two factors may increase the risk of Coronary Heart Disease more than either factor alone. This interaction

term can help capture the potential additive effect of smoking and high cholesterol.

- **Age and BMI:** As people age, their BMI may have a greater impact on their risk of Coronary Heart Disease. This interaction term can help account for the potential increased risk of Coronary Heart Disease among older individuals with high BMI.
- **Cholesterol and Dichotomous Behaviour type:** The Type A behaviour type is historically linked to increased risk of heart disease. This interaction term can help capture the potential additive effect of high cholesterol and Type A behaviour type.
- **BMI and Dichotomous Behaviour type:** Type A behaviour type is associated with stress and may interact with BMI to increase the risk of Coronary Heart Disease. This interaction term can help account for the potential combined effect of high BMI and Type A behaviour type.

We first define a base formula and then consider various interaction terms to see if they improve the model fit. After fitting the models, we compare them based on their AIC values to select the best model.

Code:

```
# Define the base formula
base_formula <- as.formula("chd69 ~ dibpat0f + age0 + cholmmol + sbp10 + bmi + smokerf + arcus0")
# Define potential interaction terms
interaction_terms <- c("age0*cholmmol", "age0:smokerf", "bmi * sbp10",
                       "cholmmol*sbp10", "age0*arcus0 ", "cholmmol:dibpat0f",
                       "smokerf * cholmmol", "age0 * bmi", "sbp10:smokerf",
                       "bmi:dibpat0f")

# Initialize a list to store models and metrics
models <- list()
metrics <- data.frame(Model = character(), AIC = numeric(), stringsAsFactors =
                        FALSE)

# Loop through interaction terms
for (i in 1:length(interaction_terms)) {
  for (j in combn(interaction_terms, i, simplify = FALSE)) {
    # Create formula with interactions
    interaction_formula <- paste(base_formula, paste(j, collapse = " + "),
                                 sep = " + ")
    full_formula <- as.formula(interaction_formula)

    # Fit the model
    model <- lrm(full_formula, data = di, x = TRUE, y = TRUE)

    # Save the model and its AIC
    models[[paste(j, collapse = ", ")]] <- model
    metrics <- rbind(metrics, data.frame(Model = paste(j, collapse = ", "),
                                         AIC = AIC(model)))
  }
}

# Sort models by AIC
metrics <- metrics[order(metrics$AIC), ]

# View the first 15 top-performing models
knitr::kable(head(metrics, 15), col.names = c("Model", "AIC"))
```

	Model	AIC
29	bmi * sbp10, age0*arcus0	1606.232
126	bmi * sbp10, age0*arcus0 , cholmmol:dibpat0f	1606.383
127	bmi * sbp10, age0*arcus0 , smokerf* cholmmol	1606.748
65	age0*cholmmol, bmi* sbp10, age0*arcus0	1606.767
331	bmi * sbp10, age0*arcus0 , cholmmol:dibpat0f, smokerf* cholmmol	1606.866
128	bmi * sbp10, age0*arcus0 , age0* bmi	1607.011
210	age0*cholmmol, bmi* sbp10, age0*arcus0 , cholmmol:dibpat0f	1607.176
332	bmi * sbp10, age0*arcus0 , cholmmol:dibpat0f, age0* bmi	1607.206
335	bmi * sbp10, age0*arcus0 , smokerf* cholmmol, age0 * bmi	1607.380
212	age0*cholmmol, bmi* sbp10, age0*arcus0 , age0* bmi	1607.424
120	bmi * sbp10, cholmmol*sbp10, age0*arcus0	1607.470
211	age0*cholmmol, bmi* sbp10, age0*arcus0 , smokerf* cholmmol	1607.523
41	age0*arcus0 , cholmmol:dibpat0f	1607.531
602	bmi * sbp10, age0*arcus0 , cholmmol:dibpat0f, smokerf* cholmmol, age0 * bmi	1607.551
5	age0*arcus0	1607.579

Explanation

After exploring various models with various combinations of interaction terms along with the full model, we went through a model selection process using AIC to compare the goodness-of-fit. We ultimately chose the model including the interactions between `bmi` and `sbp10`, and between `age0` and `arcus0` as it gave the lowest AIC on comparing with every other model combination.

- **Interaction Terms**: Interaction terms allow us to assess whether the effect of one predictor on the outcome depends on the level of another predictor. For example, the effect of BMI on CHD might vary depending on systolic blood pressure.
- **Model Comparison**: The likelihood ratio test helps determine if the addition of interaction terms significantly improves the model fit. Since the p-value is significant ($p < 0.05$), we include the interaction terms; otherwise, we would have retained the model without interactions. Coupled with the fact that the AIC was slightly better than the model without interactions though it adds a certain level of complexity given the additional number of parameters

```
# Compare using LR test
final_model <- models[[29]]
lrtest(full_model, final_model)
```

```
##
## Model 1: chd69 ~ dibpat0f + age0 + cholmmol + sbp10 + bmi + smokerf +
##     arcus0
## Model 2: chd69 ~ dibpat0f + age0 + cholmmol + sbp10 + bmi + smokerf +
##     arcus0 + bmi * sbp10 + age0 * arcus0
##
## L.R. Chisq       d.f.          P
## 7.58273819 2.00000000 0.02256469
```

3.c. Calculating Predicted Risks Once the final model is selected, we calculate the predicted probabilities of CHD for each individual in the dataset and add these predictions to the dataset.

Predicted Risks: These probabilities provide an estimate of each individual's risk of developing CHD based on the predictor values in the model. This information can be crucial for further analysis, such as assessing model calibration or making risk-based decisions.

```
di$predicted_risk <- predict(final_model, di, type="fitted")
head(di$predicted_risk)
```

```
## [1] 0.050265671 0.118150744 0.008609505 0.010966470 0.149331048 0.026458752
```

These values can be compared to the previously calculated overall risk of 0.0814838 in the cohort to see how individual risks vary based on the predictor variables.

Conclusion:

1. **Model Selection and Variable Selection**:
   - Started with a full logistic regression model.
2. **Interaction Terms**:
   - Assessed interaction effects between various predictors based on domain knowledge. Considered multiple interaction terms to improve model fit.
   - Used likelihood ratio test to compare models with and without interactions.
3. **Predicted Risks**:
   - Calculated predicted probabilities of CHD for each individual and added them to the dataset.

This approach ensures that the final model is both statistically sound and practically useful for predicting CHD risk.

Question 4: Discrimination:

4.a: AUC and ROC Curve with 95% CI

Overview:

This question focuses on evaluating the performance of the model using the AUC (Area Under the Curve) of the ROC (Receiver Operating Characteristic) curve. We will plot the ROC curve and calculate the AUC of the ROC curve, along with its 95% confidence intervals to assess how well the model can discriminate between cases and non-cases.

Code:

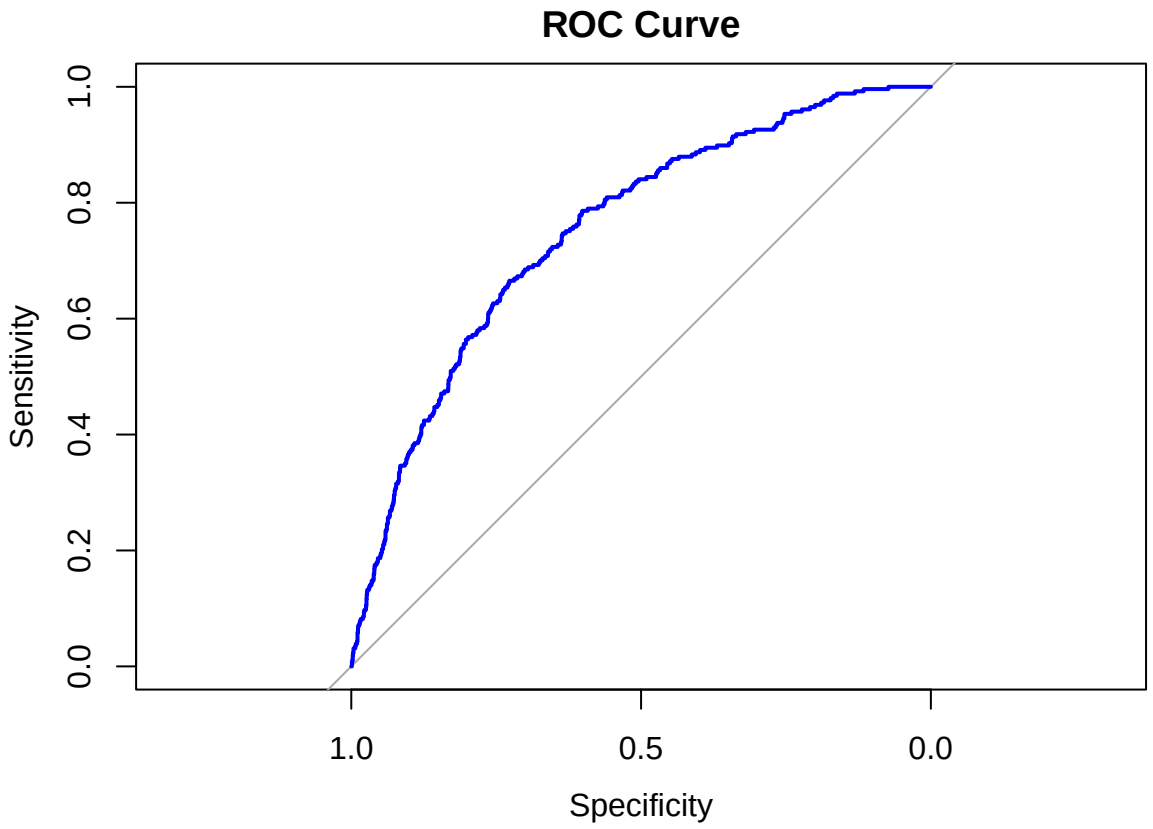
```
# Calculate the predicted probabilities using the final model (from Question 3)
di$predicted_risk <- predict(final_model, di, type = "fitted")

# ROC curve
roc_curve <- roc(di$chd69, di$predicted_risk)
```

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

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

```
# Plot ROC curve
plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
```




```
auc_value <- auc(roc_curve)
auc_ci <- ci.auc(roc_curve)
```

Output:

Table 4: AUC and 95% Confidence Interval

AUC	CI Lower	CI Upper
0.7516887	0.7218091	0.7815682

Analysis:

- The AUC of the ROC curve is 0.752, which is greater than 0.5, indicating that the model has a moderate to good ability to discriminate between cases and non-cases.
- The AUC value indicates that the model can correctly classify a randomly selected pair of individuals (one with CHD and one without) 75.2% of the time.
- The 95% confidence interval for AUC is 0.722 to 0.782, suggesting that this estimate is reasonably stable across different datasets, with no significant uncertainty in model performance.

4.b: Sensitivity and Specificity at Optimal Threshold

Overview:

This question focuses on finding the optimal threshold that maximizes the sum of sensitivity and specificity. We will plot the ROC curve, find the threshold, and report both sensitivity and specificity at that threshold.

Sensitivity: Sensitivity (True Positive Rate) measures the proportion of actual positive cases that are correctly identified by the model.

Specificity: Specificity (True Negative Rate) measures the proportion of actual negative cases that are correctly identified by the model.

Approach:

- Find the threshold that maximizes sensitivity + specificity.
- Report sensitivity and specificity at the optimal threshold.

Code:

Extract the numerical values of the optimal threshold, sensitivity, and specificity at that threshold from the ROC curve.

```
# Find the threshold that maximizes sensitivity + specificity
coords <- coords(roc_curve, "best", ret = c("threshold", "sensitivity", "specificity"))

# Report sensitivity and specificity at the optimal threshold
optimal_threshold <- coords$threshold
sensitivity_at_threshold <- coords$sensitivity
specificity_at_threshold <- coords$specificity
```

Output:

Value of Optimal Threshold, Sensitivity and Specificity at Threshold:

```
## [1] "Optimal Threshold:  0.094"
```

```
## [1] "Sensitivity at Threshold:  0.665"
```

```
## [1] "Specificity at Threshold:  0.727"
```

Analysis:

The optimal threshold is 0.094, where the sum of sensitivity and specificity is maximized. At this threshold:

The sensitivity (True Positive Rate):

- This means that 66.5% of individuals with Coronary Heart Disease (CHD) are correctly identified by the model.

The specificity (True Negative Rate):

- This means that 72.7% of individuals without CHD are correctly identified by the model.

This threshold ensures a good balance between sensitivity and specificity, though further analysis of model performance at different thresholds may be beneficial.

4.c: Adjusted AUC using Bootstrap Validation

Overview:

In this case, the AUC was adjusted using the bootstrapping method to correct for possible optimism bias in the model. By resampling the training data 200 times, the adjusted AUC is calculated and compared with the original AUC.

Approach:

- Use the `validate` function from the `rms` package to adjust the AUC using the bootstrap method.
- The AUC is optimised by bootstrapping the data 200 times and comparing the original AUC with the adjusted AUC.

- The AUC is extracted from the Somers' Dxy index, and the adjusted AUC is calculated.
- Compare the original AUC with the adjusted AUC to assess optimism bias correction.

Code:

```
# AUC adjustment using validate function (bootstrap method)
validation_result <- validate(final_model, method = "boot", B = 200)

# View the structure of the validation_result
validation_result
```

```
##           index.orig training    test optimism index.corrected   n
## Dxy           0.5034   0.5146  0.4953   0.0193          0.4841 200
## R2            0.1390   0.1466  0.1332   0.0134          0.1256 200
## Intercept     0.0000   0.0000 -0.1061   0.1061         -0.1061 200
## Slope         1.0000   1.0000  0.9469   0.0531          0.9469 200
## Emax          0.0000   0.0000  0.0327   0.0327          0.0327 200
## D             0.0615   0.0650  0.0589   0.0061          0.0554 200
## U            -0.0006  -0.0006  0.0002  -0.0008          0.0002 200
## Q             0.0621   0.0656  0.0587   0.0069          0.0552 200
## B             0.0699   0.0693  0.0703  -0.0009          0.0708 200
## g             1.1018   1.1367  1.0711   0.0656          1.0362 200
## gp            0.0737   0.0752  0.0721   0.0031          0.0706 200
```

```
# Extract the original AUC (index.orig) and the adjusted AUC (index.corrected)
index_orig <- validation_result[1, "index.orig"]   # Somers' Dxy
index_corrected <- validation_result[1, "index.corrected"]
# Optimism-corrected Somers' Dxy

original_AUC <- 0.5*(index_orig + 1)  # Calculate the original AUC
adjusted_AUC <- 0.5*(index_corrected + 1)  # Calculate the adjusted AUC
```

Output:

Unadjusted AUC, adjusted AUC from the validate method:

Table 5: Comparison of Original and Adjusted AUC

Original AUC	Adjusted AUC
0.7516887	0.7420478

Analysis:

- The original AUC of the model was 0.752, indicating a moderate to good ability to discriminate between cases and non-cases of Coronary Heart Disease (CHD).
- The adjusted AUC, after correcting for optimism bias using the bootstrap method, was 0.742.
- This value provides a more realistic estimate of the model's performance on unseen data by accounting for potential overfitting in the original AUC calculation.

4.d: 10-Fold Cross-Validation for Adjusted AUC

Overview:

In this task, we perform a 10-fold cross-validation on a logistic regression model to estimate the adjusted AUC and compare it with the unadjusted AUC values. Cross-validation helps in assessing the model's

generalization ability by training and testing the model on different subsets of the data.

Cross-Validation: Cross Validation is a resampling technique used to evaluate ML models by training and testing on multiple subsets of the data. It helps in estimating the model's performance on unseen data and reducing overfitting.

Code:

```
set.seed(154550)
# Set the number of folds for cross-validation
num_folds <- 10

# Create a 10-fold cross-validation partition
folds <- createFolds(di$chd69, k = num_folds, list = TRUE)

# Store the AUC values
auc_values <- c()

# Perform 10-fold cross-validation
for (i in 1:num_folds) {
  # Define training and testing sets
  train_data <- di[folds[[i]], ]
  test_data <- di[-folds[[i]], ]

  # Fit the logistic model on the training data
  model_cv <- lrm(chd69 ~ dibpat0f + age0 + cholmmol + sbp10 + bmi + smokerf
                  + arcus0 + bmi * sbp10 + age0*arcus0 , data = train_data,
                  x = TRUE, y = TRUE)

  # Predict on the test set
  predicted_prob <- predict(model_cv, test_data, type = "fitted")

  # Calculate the AUC for the current fold
  roc_curve_cv <- roc(test_data$chd69, predicted_prob)
  auc_values[i] <- auc(roc_curve_cv)
}

# Calculate the average AUC from the cross-validation
avg_auc <- mean(auc_values)
ci_auc <- quantile(auc_values, probs = c(0.025, 0.975))
```

Output:

Table 6: Comparison of Cross-Validated, Unadjusted and Adjusted AUC

Cross-Validated AUC	Unadjusted AUC	Adjusted AUC	CI Lower	CI Upper
0.7003599	0.7516887	0.7420478	0.6703027	0.7290904

Analysis:

- Cross-validated AUC: The 10-fold cross-validation resulted in an average AUC of 0.7 with a 95% confidence interval ranging from 0.67 to 0.729. This value reflects the model's performance on unseen data and provides an estimate of its generalization ability.

- Unadjusted AUC: The unadjusted AUC was calculated as 0.752. This AUC was computed using the entire dataset without cross-validation, and it tends to be optimistic due to overfitting to the data.
- Adjusted AUC: The adjusted AUC, after correcting for optimism bias using the bootstrap method, was 0.743. This value provides a more realistic estimate of the model's performance on unseen data by accounting for potential overfitting in the original AUC calculation.
- Comparison: The cross-validated AUC is lower than the unadjusted and adjusted AUC, which suggests that the unadjusted model might be overfitting to the training data. The cross-validation process, by testing the model on different folds, provides a more conservative estimate of the model's performance.

Question 5: Calibration

5.a: Calibration Curve and Slope-Intercept

Overview:

To evaluate the model's calibration, we will plot the calibration curve using the rms package in R. The slope and intercept of the calibration curve will also be reported, reflecting how well the model's predictions align with the observed data.

Calibration: - Calibration assesses how well the predicted probabilities from the model match the observed outcomes. A well-calibrated model has predicted probabilities that closely align with the true probabilities of the outcome. The calibration formula for binary outcome variable is given by:

$logit(Y = 1) = a + b \times logit(\hat{Y})$

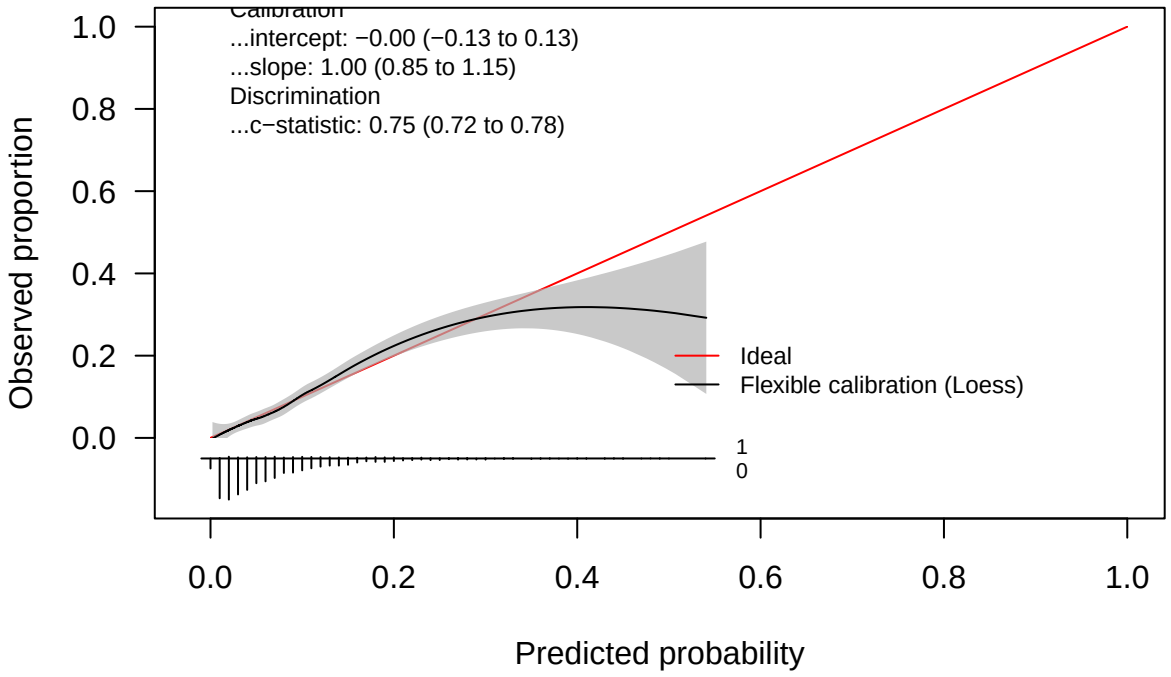
where Y is the observed outcome, $\hat{Y}$ is the predicted probability, and a and b are the intercept and slope of the calibration curve, respectively.

Approach:

- Plot the calibration curve and calculate the slope and intercept of the calibration curve by using the `val.prob.ci.2` function from the `CalibrationCurves` package.
- The function provides the slope, and intercept and c-statistic of the calibration curve by comparing the predicted probabilities with the observed outcomes using logistic regression.

Code:

```
# Plot the calibration curve
CalibrationCurves::val.prob.ci.2(di$predicted_risk, di$chd69)
```




```
## Call:
## CalibrationCurves::val.prob.ci.2(p = di$predicted_risk, y = di$chd69)
##
## A 95% confidence interval is given for the calibration intercept, calibration slope and c-statistic.
##
##           Dxy       C (ROC)            R2              D      D:Chi-sq
##  5.033101e-01  7.516551e-01  1.389504e-01  6.151295e-02  1.950118e+02
##           D:p             U      U:Chi-sq           U:p             Q
##  0.000000e+00 -6.341154e-04  1.136868e-12  1.000000e+00  6.214706e-02
##         Brier     Intercept         Slope          Emax  Brier scaled
##  6.985139e-02 -8.552357e-13  1.000000e+00  3.600453e-13  6.670952e-02
##          Eavg           ECI
##  6.837367e-03  2.241619e-02
```

The intercept: -0.00 (-0.13 to 0.13)

The slope: 1.00 (0.85 to 1.15)

C-statistic: 0.75 (0.72 to 0.78)

Analysis:

- The intercept and slope of the calibration curve are close to 0 and 1, respectively, indicating that the model's predicted probabilities align well with the observed outcomes.
- The C-statistic (concordance index) of 0.75 suggests that the model has good discrimination ability, with a 75% probability of correctly ranking a randomly selected pair of individuals (one with CHD and one without) based on their predicted probabilities.
- Also, we notice that the C-statistic is the same as the AUC value because it indicates the same thing, which is the model's ability to discriminate between positive and negative outcomes.

5.b: Hosmer-Lemeshow Test

Overview:

Estimate the goodness of fit by the method of Hosmer and Lemeshow and interpret the test results.

Hosmer-Lemeshow Test: It is applied to assess the goodness of fit of the logistic regression model. This test divides the data into deciles (one-tenths) based on predicted probabilities and evaluates whether the observed outcomes match the predicted values within these groups.

- *Null Hypothesis:* There is no statistical significance between the observed and predicted outcomes, indicating a good fit.
- *Alternative Hypothesis:* There is statistical significance between the observed and predicted outcomes, indicating a poor fit.

Approach:

- Perform the Hosmer-Lemeshow test using the `hoslem.test` function from the `ResourceSelection` package.
- The test compares the observed and predicted outcomes within deciles and provides a chi-squared statistic, degrees of freedom, and p-value to assess the model's goodness of fit.
- p <= 0.05 : Reject the null hypothesis, indicating a poor fit.
- p > 0.05 : Fail to reject the null hypothesis, indicating a good fit.

Code:

```
library(ResourceSelection)
```

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

```
hl_test = hoslem.test(di$chd69, di$predicted_risk)
```

Output:

The result from the Hosmer-Lemeshow test:

```
## Hosmer-Lemeshow Test: Chi-squared = 4.651965 , df = 8 , p-value = 0.794044
```

Analysis:

- We fail to reject Null Hypothesis since the p-value is 0.794, larger than 0.05, which means that there is no statistical significance between the observed and predicted outcomes, indicating a well calibrated model.
- This suggests that we were able to predicy the Coronary Heart Disease (CHD) outcomes well using the logistic regression model.

5.c: Prediction Model with Agegroup

Overview:

A new logistic regression model will be created using only the variable agegroup as the predictor. The discrimination ability of this simplified model will be estimated using the Area Under the Curve (AUC), a measure of the model's ability to differentiate between positive and negative outcomes.

Code:

```
# Assuming 'agegroup' is a variable in the dataset
agegroup_model <- lrm(chd69 ~ agegroup, data = di)

# Predict probabilities for the agegroup model
```

```r
agegroup_probs <- predict(agegroup_model, data = di, type = "fitted")

# Compute the ROC curve and AUC for the agegroup model
roc_agegroup <- roc(di$chd69, agegroup_probs)
```

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

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

Output:

The AUC value of the new Agegroup model.

```r
auc(roc_agegroup)
```

```
## Area under the curve: 0.6063
```

Analysis:

- The AUC value of 0.6063 indicates this agegroup model can provide some useful predictions with only agegroup as a predictor.
- This suggests that agegroup alone has some ability to differentiate between positive and negative outcomes of the Coronary Heart Disease (CHD).

5.d: ROC Test

Overview:

The discrimination performance of the original model and the simplified agegroup-only model will be compared statistically. The DeLong test will be employed to determine if the difference in the AUCs of the two models is statistically significant. The test result will help assess whether including additional predictors improves the model's performance.

Null Hypothesis - There is no discrimination difference between the original model and the agegroup-only model. **Alternate Hypothesis** - There is a significant discrimination difference between the original model and the agegroup-only model.

- p-value <= 0.05 : Reject the Null Hypothesis, indicating there's a significant difference in the discrimination.
- p-value > 0.05 : Fail to reject Null Hypothesis, indicating there's no significant difference in the model's discrimination.

Code:

```r
# Get the roc curve from question 4
roc_final_model <- roc_curve

# DeLong test for comparing AUCs
roc_comparison <- roc.test(roc_final_model, roc_agegroup)
roc_comparison
```

```
##
##  DeLong's test for two correlated ROC curves
##
## data:  roc_final_model and roc_agegroup
## Z = 8.544, p-value < 2.2e-16
## alternative hypothesis: true difference in AUC is not equal to 0
```

```
## 95 percent confidence interval:
##  0.1120352 0.1787378
## sample estimates:
## AUC of roc1 AUC of roc2
##   0.7516887   0.6063021
```

Analysis:

- The p-value is less than 0.05 indicating that there is a significant difference in the discrimination performance between the original model and the agegroup-only model.
- This suggests that agegroup alone is not sufficient to achieve the same level of discrimination as the original model, which includes additional predictors.

5.e: ROC Curves Comparison

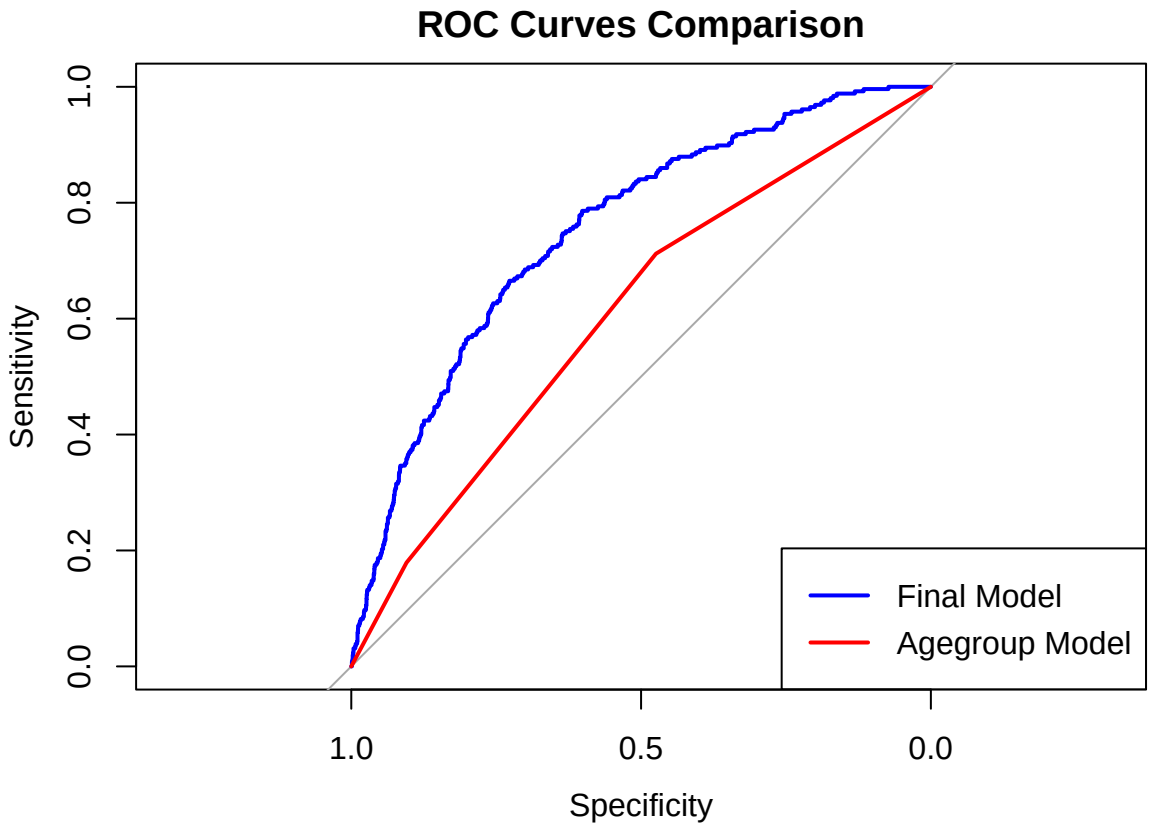
Overview:

The Receiver Operating Characteristic (ROC) curves for both the original model and the simplified agegroup-only model will be plotted on the same graph. This visualization will allow for a direct comparison of the two models' discrimination capabilities. Key features such as AUC values and curve shapes will be highlighted.

Code:

```r
# Plot both ROC curves together
plot(roc_final_model, col = "blue", main = "ROC Curves Comparison", lwd = 2)
lines(roc_agegroup, col = "red", lwd = 2)

# Add a legend
legend("bottomright", legend = c("Final Model", "Agegroup Model"),
       col = c("blue", "red"), lwd = 2)
```



Analysis:

- The ROC curve for the final model (blue) is closer to the top-left corner, indicating better discrimination performance compared to the agegroup-only model (red).
- This indicates that the additional predictors in the final model contribute to its improved performance compared to a model based solely on agegroup and that agegroup model is not strong enough for high-confidence decision-making.
- However, the AUC values of both the models are greater than 0.5, indicating that both models have some ability to discriminate between the cases and non-cases od Coronary Heart Disease (CHD).

Question 6

6.a: Decision Curve Analysis

Overview:

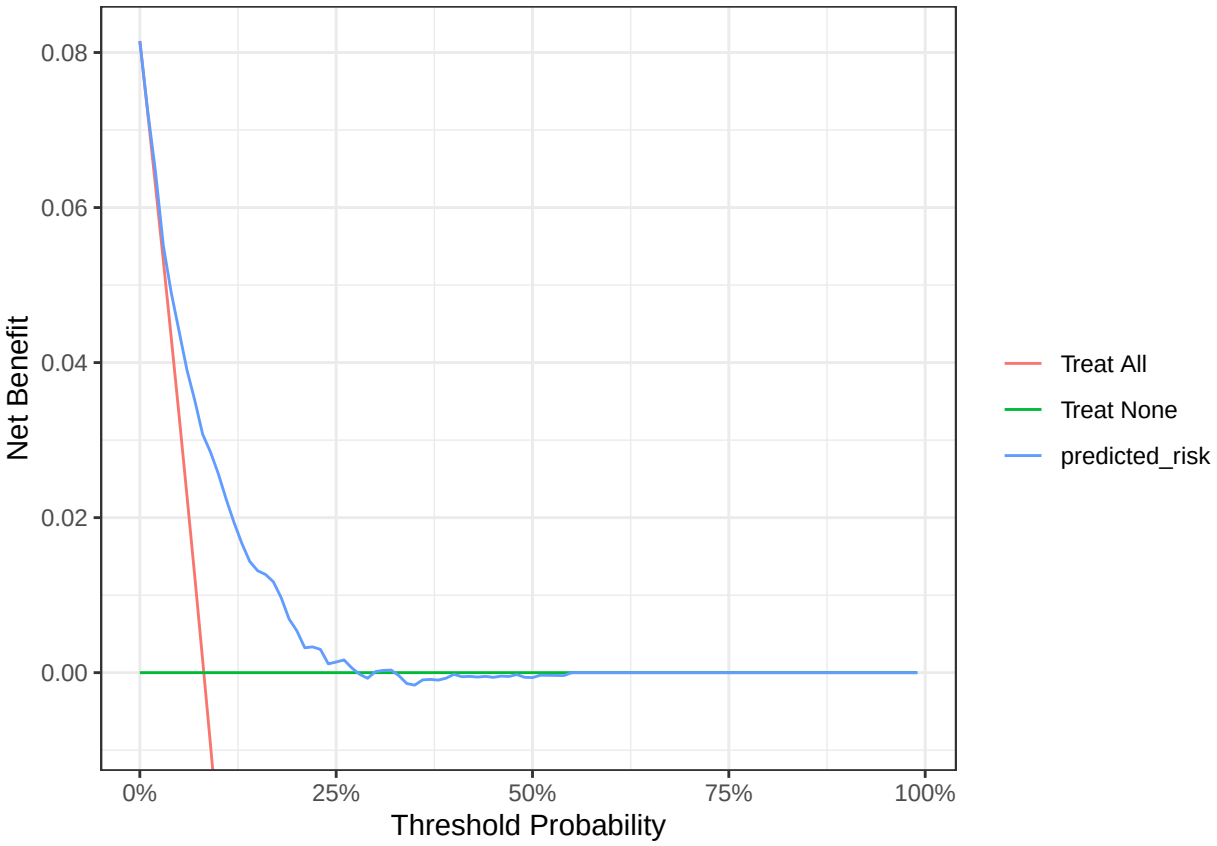
Decision Curve Analysis is used to evaluate the clinical benefits of the model by comparing the net benefit of the model with that of the default strategies, that is "Treat-all" and "Treat-none". These strategies are based on the disease prevalence in our study population.

Code:

```
dca_data <- data.frame(predicted_risk = di$predicted_risk, actual_label = di$chd69)
dca_results <- dca(actual_label ~ predicted_risk, dca_data)
```

Output:

The Decision Curve Analysis plot, with the default strategies and our chosen model (predicted_risk):



Analysis:

In the threshold range approx. 0-3% the model goes in line with the "treat-all" strategy. This is expected as at very low thresholds the cost of missing true positive is high and outweighs the cost of unnecessary treatments, making "treat-all" strategy useful (e.g. for serious diseases or when the cost of treatment is low).

Then in range 3-25% the predicted risk model's net benefit surpasses the "treat-all" strategy, staying constantly above the default strategies. This suggests its clinical usefulness in this threshold, as it means the model strikes a good balance of the trade-off between correctly identifying patients at risk and avoiding unnecessary treatments.

Looking at the entire plot we can observe that the model's net benefit converges to "treat-none" strategy's net benefit of 0, as threshold probability increases. However it's worth to note that the model's net benefit oscilates around 0, often going below the "treat-none" strategy in the range 25-50%.

6.b: Clinical Usefulness

Analysis:

To access whether the model is clinically useful or harmful we need to analyze the relation of its net benefit to the default strategies, namely it should be higher for the given threshold.

As stated above, the model doesn't seem to make a difference at very low thresholds, what is expected. Then the net benefit quickly diverges and becomes larger in 3-25% threshold probability range. This is the range where the model is the most clinically useful.

In general for higher thresholds we would expect the cost of treatment being relatively high. A model that misjudges the case could lead to unnecessary interventions and causing more harm than not using it at all.

Thus the range 25-50% is a range where the model doesn't provide value and is possibly harmful. Beyond this range the model's net benefit is 0, indicating no practical benefits.

To sum up, the presented Decision Curve Analysis plot indicates that our model is clinically useful in the range approx. 3-25%.

6.c: Net benefit comparison with age-only model.

Overview:

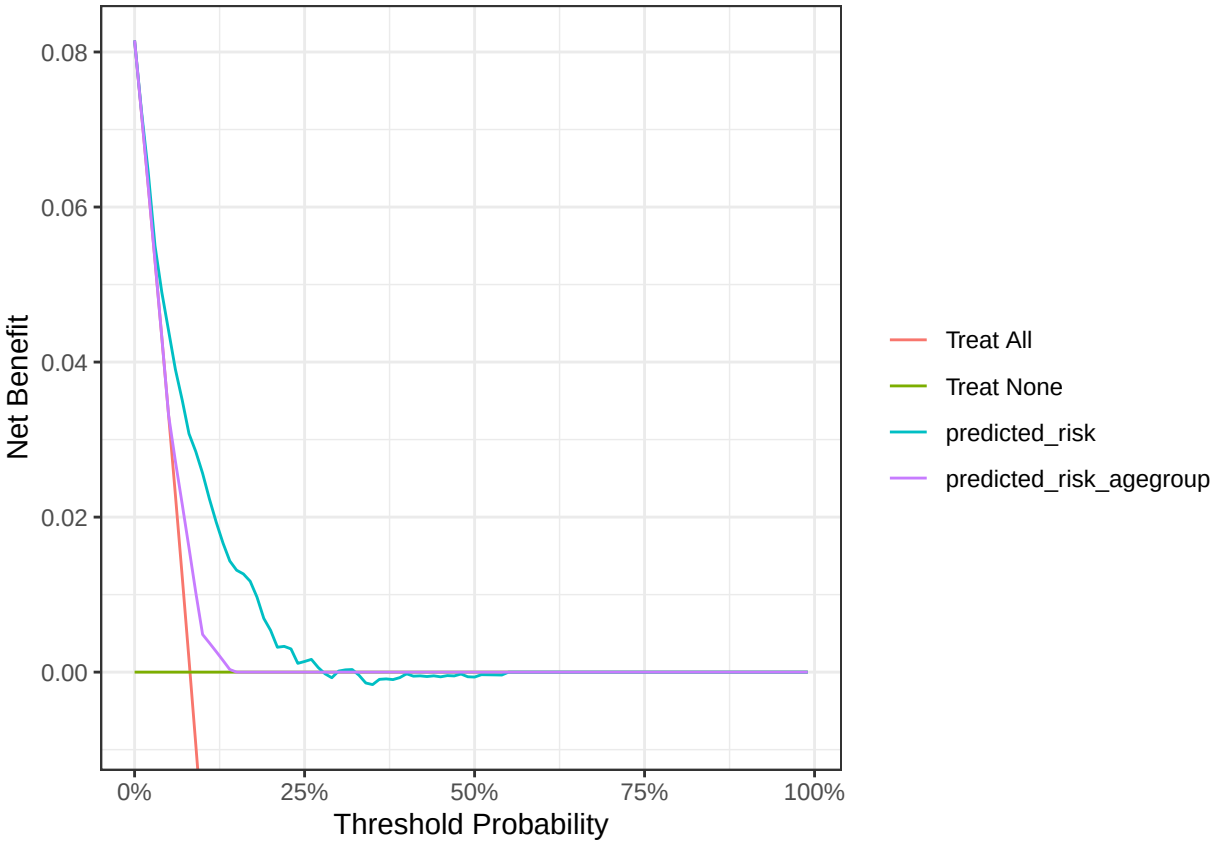
5.c is a model only using variable agegroup as a predictor. The comparison between the two plots allows us to assess which model provides a better clinical performance across different threshold probabilities. The expectation is that the agegroup model is going to be less useful, as we found a statistically significant argument for using the more complex model in the previous exercises.

Code:

```
dca_data_combined <- data.frame(predicted_risk_agegroup = agegroup_probs,
                                predicted_risk = di$predicted_risk,
                                actual_label = di$chd69)
dca_combined <- dca(actual_label ~ predicted_risk + predicted_risk_agegroup,
                    dca_data_combined)
```

Output:

The Decision Curve Analysis plot with default strategies like before, "predicted" risk showing net benefit of our final model and "predicted_risk_agegroup" showing the net benefit of the simplified age only model:



Analysis:

As expected, the final model performs better than the one based solely on age. Comparing the net benefit curves of both, we can see that the age model shows clinical benefits in a way smaller range of threshold probabilities, as well as its net benefit value is visibly way below the final model across the entirety of this range.

To conclude it is only slightly more useful than the default strategies and less useful than the final model.

Question 7

Discussion on clinical usefulness of the final model:

Using various metrics, we find consistent results suggesting the usefulness of the chosen model. In Exercise 6, we used Decision Curve Analysis to discuss the clinical benefits in particular. We found that the model is helpful within the range of approximately 3-25%. It may not be advisable to use it in the range of 25-50%, and it does not offer real benefits in the remaining ranges.

In practice, such a model appears to be clinically useful, as it adds value within threshold ranges typically used when dealing with serious diseases. In such cases, the consequences of missing a true positive case are very high, while treatments may also be invasive or costly. Therefore, a model that shows clear benefits and provides better assessments than default strategies is desirable, enabling more accurate identification of cases.

Summary of the model's performance:

We have developed a model and performed various standard tests to measure its performance. Our model has an AUC of 0.7 with a 95% confidence interval ranging from 0.67 to 0.729, as a conservative estimate indicating model's usefulness compared to classifying patients at random. Additionally, we assessed the potential for recalibration and performed Decision Curve Analysis to identify the clinical settings where the model is most applicable. These steps further convinced us of the model's usefulness.

Possible next steps:

Assuming our goal is CHD prediction in patients, the next steps as researchers should include a comparison with state-of-the-art methods. If our model is found to be comparable, we shall proceed with external validation, that is look for a different similar datasets and use these to further test the model. Depending on our goals, we may also need to assess its transportability, i.e., its performance on datasets with population different from ours. If these steps are successful, we could proceed to clinical testing, such as through a randomized trial.