# Compulsory exercise: Team SuperGreat

MA8701 Advanced Statistical Learning V2023

Nora Aasen, Elias Angelsen, Jonas Nordstrom

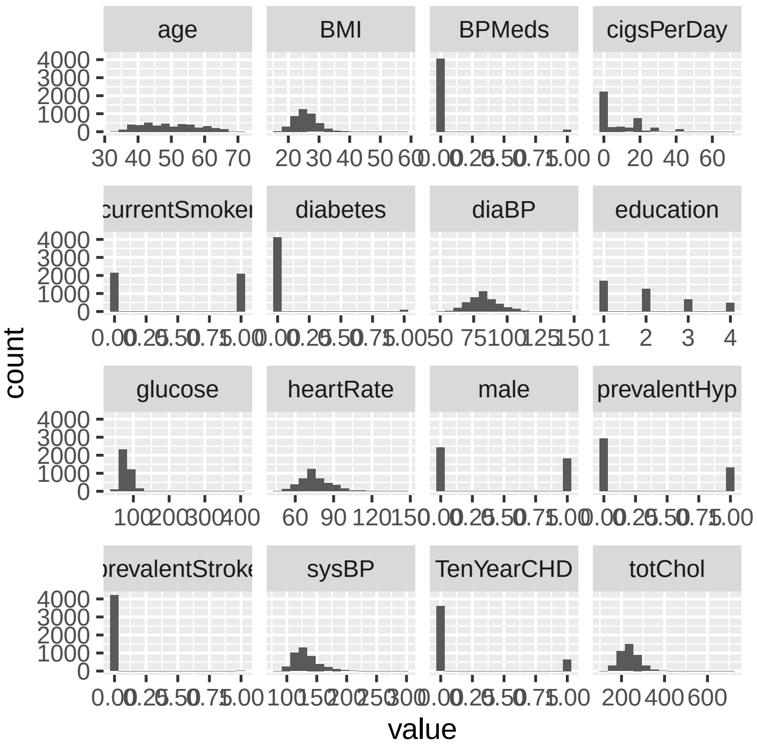
15 mars, 2023

## Introduction

In this project we have studied the Framingham Coronary Heart Disease Dataset. This dataset contains patient information for inhabitants in Framingham, Massachusetts, and is typically used to predict the chance of getting coronary heart disease (CHD) within the next 10 years. For this project, however, we intend to use lasso to find the most important risk factors. This is our main objective, and to see that the variables really are significant, we fit a logistic model using the variables we have selected. A secondary, although large, part of our project, is to explore methods for imputing missing data. We will do single regression imputation manually and through the 'mice' package, and investigate a bit what the Lasso is doing on the imputed data sets, compared to the complete case.

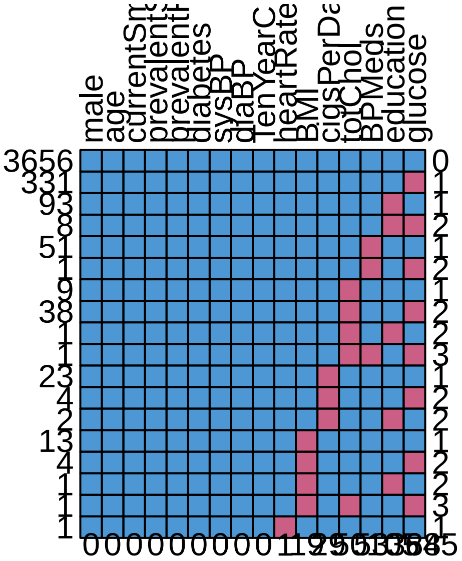
We start by examining the dataset.

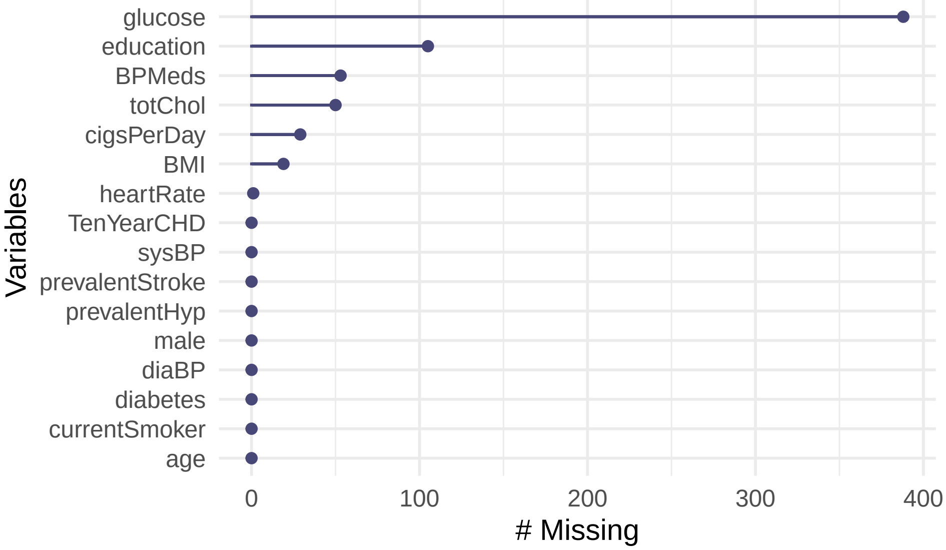
### Exploratory Analysis



This data set contains 4238 observations, 15 covariates and a binary response variable `TenYearCHD`. We will try to fit a logistic regression model. The response variable has 644 observations that are 1, which equals about 15.2% of the total observations. Most of our covariates are either binary, or numeric. However, we notice that the variable education is most likely a categorical covariate. We could not find any further elaboration for which four categories the numbers represent, so based on the frequency of each value and qualified guessing, we changed it to a factor variable and defined the four categories as none, hs, college, post-grad.

The next thing we looked at was the number of missing data in our data set.

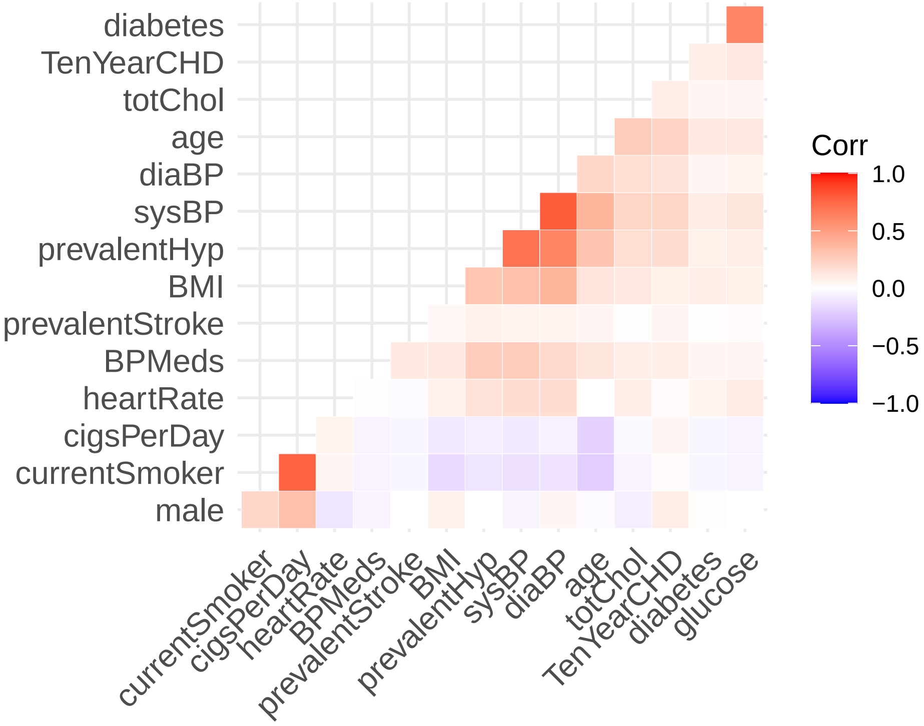




As we can see there are six covariates that has missing data: `glucose`, `education`, `BPMeds`, `totChol`, `cigsPerDay`, and `BMI`. We cannot use the rows that contain missing values as is. The easiest solution is to remove all rows that contains `NA`'s. This is the *complete case* solution. We split the data into training and test sets, as well as copying the complete case data set for later solutions.

The complete data set contains 3656 observations and the response variable has 557 observations that are 1, which equals about 15.2% of the total observations. As we can see, the proportion of positive observations in the response is the same, which is a good indicator that our data is missing at random (MAR), as we will discuss later.

To end our exploratory analysis, we make a correlation plot of the complete data set, but note that we have removed the multi-class variable ‘education’.



The above tells us that some variables are highly correlated. These are ‘currentSmoker’ and ‘cigsPerDay’, ‘sysBP’ and ‘diaBP’, ‘prevalentHyp and ’diaBP’/‘sysBP’, and ‘diabetes’ and ‘glucose’. Initially, we therefore expect at most one of ‘currentSmoker’ and ‘cigsPerDay’, one of ‘sysBP’ and ‘diaBP’, and one of ‘diabetes’ and ‘glucose’ to be picked out by the Lasso. The case for ‘prevalentHyp’ is slightly more complex. It’s influence towards the response may be absorbed into ‘sysBP’ and/or ‘diaBP’, in the eyes of the Lasso. It could also dominate both and push both of these out of the model, or it could be included in the model, but with much lower significance than the other included variables. We don’t know yet.

From the exploratory analysis we know that there are many missing data points. However, the complete case data set is quite large, enabling the use of purely complete cases. Our main focus for this project will partially be to compare the results from doing lasso on the complete case with the results from doing lasso on an imputed data set(s).

## Missing Data

In our data set, we have missing values, and we are additionally going to handle these missing values in a slightly more refined manner than just considering the complete case.

Recall that there are several types of mechanisms for missing data. Let $Z = (X, y)$ denote the full collection of covariates and responses, respectively, and we let a subscript mis/obs indicate whether we are restricting $Z$ (or $X$) to the missing or observed parts, respectively. We may form an indicator (0-1) matrix $R$ indicating missing (0) and observed (1) covariates. Assume $\psi$ is short for the parameters in the distribution of $R$.

The missing data may be characterized by the conditioning in the distribution of $R$. We define the data to be:

- missing completely at random (MCAR) if $P(R|Z, \psi) = P(R|\psi)$,
- missing at random (MAR) if $P(R|Z, \psi) = P(R|Z_{obs}, \psi)$,
- missing not at random (MNAR) if $P(R|Z, \psi) = P(R|Z, \psi)$ (i.e. we don't have MCAR or MAR).

By for example exploring the missing pattern of the variable 'cigsPerDay', we obtain a clear indication that our missing mechanism is not MCAR. No non-smoker has failed to answer the question "how may cigarettes do you smoke a day?", which is a question only aimed at smokers. The simple explanation may be that the survey they answered automatically fills in 0 for 'cigsPerDay' if you claim to be a non-smoker. In more mathematical terms, the missingness of 'cigsPerDay' depends on the observed answer to "do you smoke?" (found in variable 'currentSmoker'), indicating that we do not work with MCAR data. Luckily, most methods are applicable if our missingness is at least MAR.

We will assume that the missing mechanism is MAR, as there is no clear reason to suspect it to be MNAR. An example of a pressing obstruction to being MAR can again be found in 'cigsPerDay'. In the real world, if smokers have failed to report 'cigsPerDay', it may for example be because they smoke so much that they are ashamed to answer the question (and skips it), but we will simply assume that such a thing is not happening, as we trust people to answer truthfully (often, at least) if they volunteer for medical studies.

To treat the missing data, we will use single imputation, as multiple imputation may cause difficulties with the resulting inference, as Rubins rules needs to be combined with the Lasso, bootstrap and concluding inference. Multiple imputation has therefore not been the focus in this project.

The single imputation technique we will use is simply regression imputation, where we adapt our regression technique depending on the type of variable imputed. For continuous variables, we simply use a linear regression model. For binary variables, we use logistic regression to classify their values, and for the variable "education", which is a four-class variable, we have utilized kNN for multiclass imputation. This is implemented manually, but this could have been done using the 'MICE' package and the function 'mice'.

To avoid encountering observations with more than one missing value, and hence problems with regressing, we remove all samples with more than one NA.

Our data is split into training and test sets, with the test-to-training ratio being 'r tr' of the original dataset. In order to avoid data leakage in our imputation of the test set, we fit the imputation models on the training set. The main idea is that the test set should be viewed as several independent observations. Using the test set to impute itself will use information not present at the time of training and will yield unintended correlation. This is again done manually, but could also have been done using 'mice.reuse', as we will do later.

Note that we do not include the response (TenYearCHD) in the regression, and in order to avoid too much correlation between the imputed samples, we always base the regression models on the complete case data, instead of letting the imputed values for variable $n$ regress to impute variable $n + 1$.

First, we remove all samples with more than one covariate missing, as this is only 61 samples.

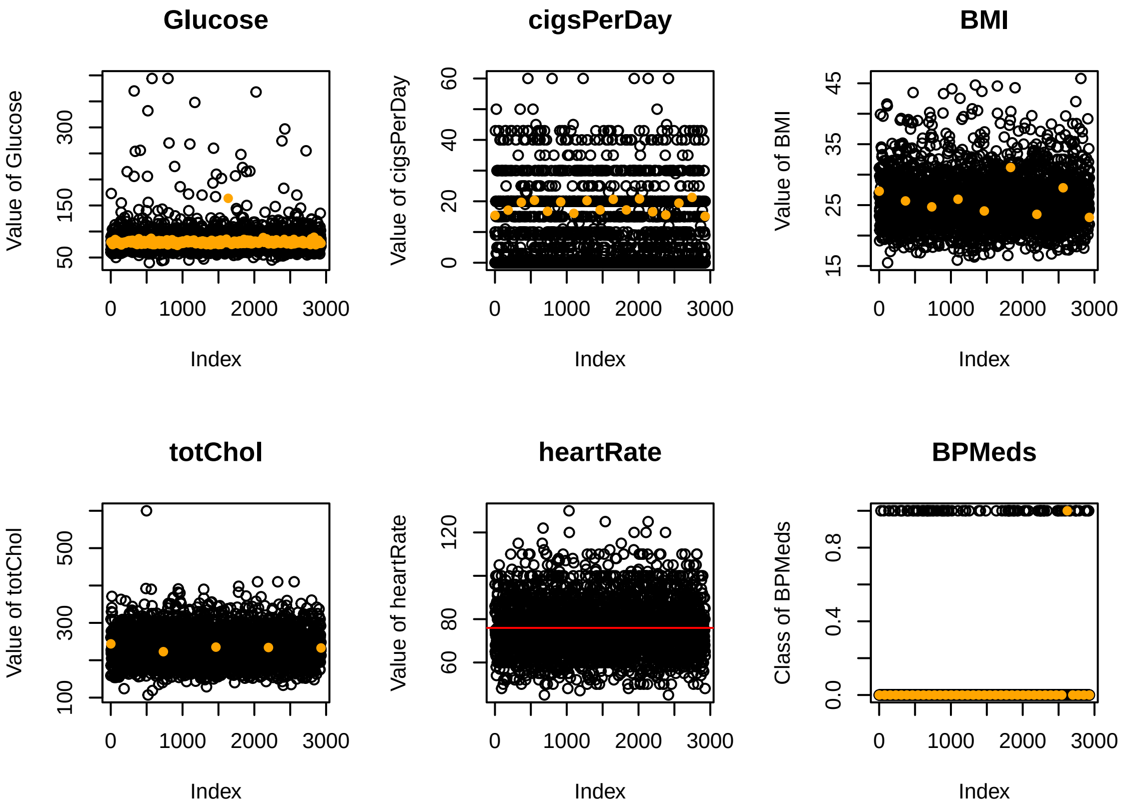
We further split the data into further training and test sets with and without the response 'TenYearCHD', following the training-test-ratio that we have previously set.

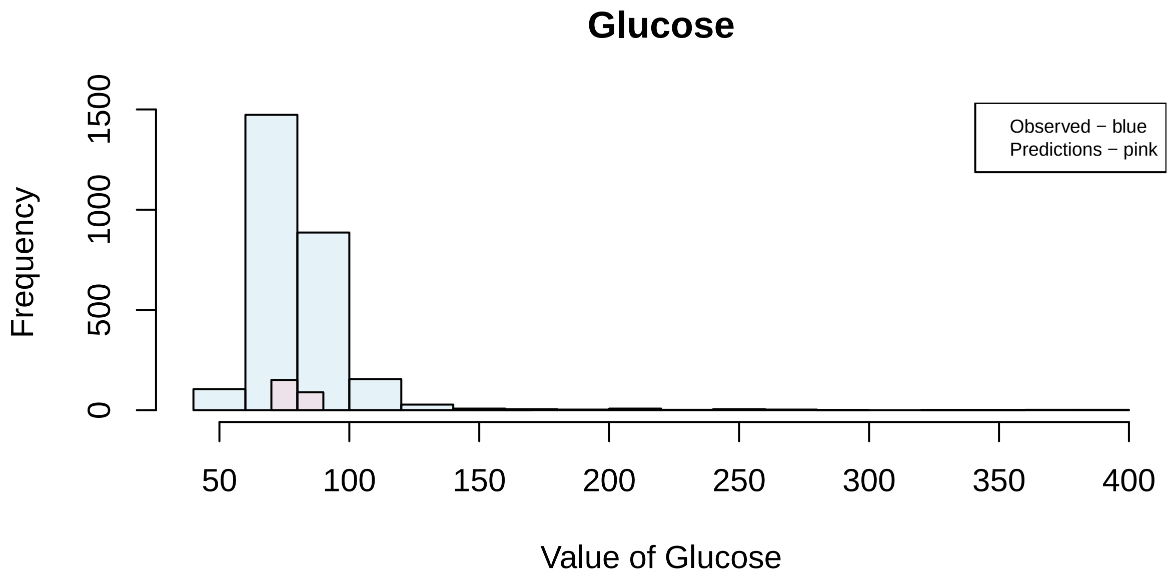
Using these data sets, we make regression models for each missing variable based on the data we have. These models automatically neglects NA's. For 'glucose', 'cigsPerDay', 'BMI', 'totChol' and 'heartRate', we fit linear models, while for the binary variable ‘BPMeds’, a logistic model is fit. The multi-class variable ‘education’ will be imputed using a kNN-model.

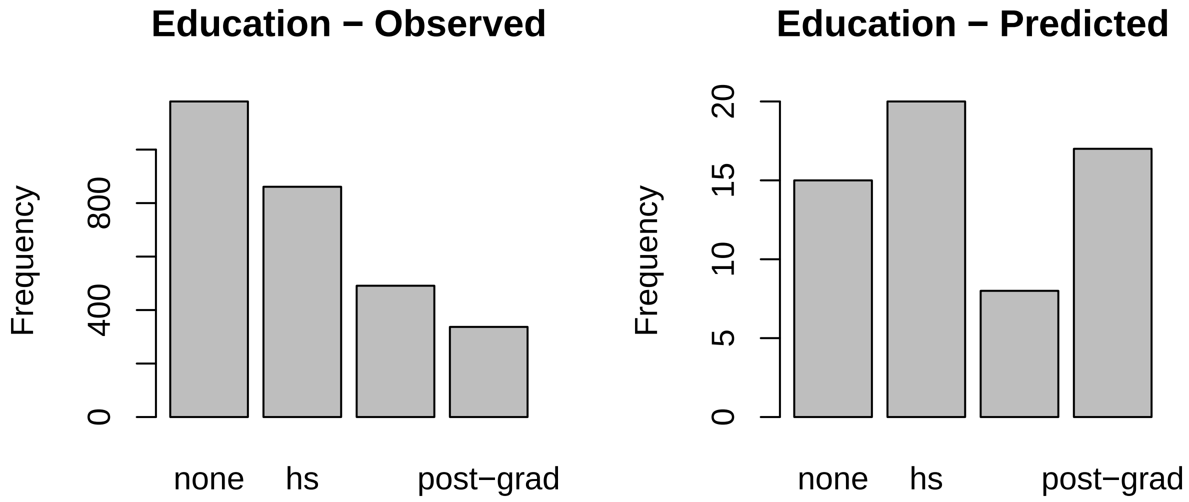
To predict the missing values for each variable, we pick out those samples with missing values of each variable from the training and test set. Note that we are imputing for both the training and test set, even though the sampled (therefore “random”) split into training and test set may have sorted all NA’s of a variable (e.g. heartRate) into a single set (i.e. into either training or test set). This is not a problem, as we are only picking out the incomplete cases. Therefore, predicting and filling in the missing values in the complete part is a vacuous procedure, not yielding any problems.

We then predict the missing values for imputation. For the logistic model fit to predict ‘BPMeds’, we assign a prediction class 1 or 0 depending on whether the predicted value is above or below 0.5.

To see how our predictions compare to the complete case, we plot (in different ways) the predicted values/classes. For illustrating different types of plots useful for this, we plot glucose twice, as both a transparent histogram and a point plot. This is only for illustrational purposes and therefore only done for the training data with its imputed values.







For most of the linear predictors, we are imputing using linear models. As we can see in these plots, we are close to the mean, but we are including more variance than what mean imputation would be able to. The binary variable 'BPMeds' is often classified to zero, which is expected, as the variable explains whether or not the patient was taking blood pressure medicine at the time of the survey. The 'education' variable is predicted using kNN, and we predict surprisingly many with higher education. This can imply that the kNN-model we fit is not optimal for predicting 'education'.

Lastly, we update our data with the newly imputed values.

We have obtained completely imputed data sets. To see that our procedure has worked, we could consider the missing data patterns of the newly constructed data sets.

We have managed to impute the missing values, but there are several steps in this procedure that could be improved.

First of all, we could have used more flexible models for imputation than linear imputation, and we could have fit several different models on the training set and done evaluation procedures within the training set (e.g. cross validated optimization of ROC-AUC) to pick the models before fixing a model to impute each variable.

Some variables, such as 'cigsPerDay', are in some sense discrete, although we have treated them as continuous (as it is possible to smoke e.g. $2,73$ cigarettes per day). These could have been rounded off to integers, but we didn't see why this would be necessary. We did not include the response ('TenYearCHD') in the regressions. It is not clear to us why it would be a better choice to include it, as we don't want the imputed data to be overfitted towards the response. More reading and testing would be needed to figure out the optimal solution, if there is any canonical choice, and it would be interesting to see how our results changed if the response were included.

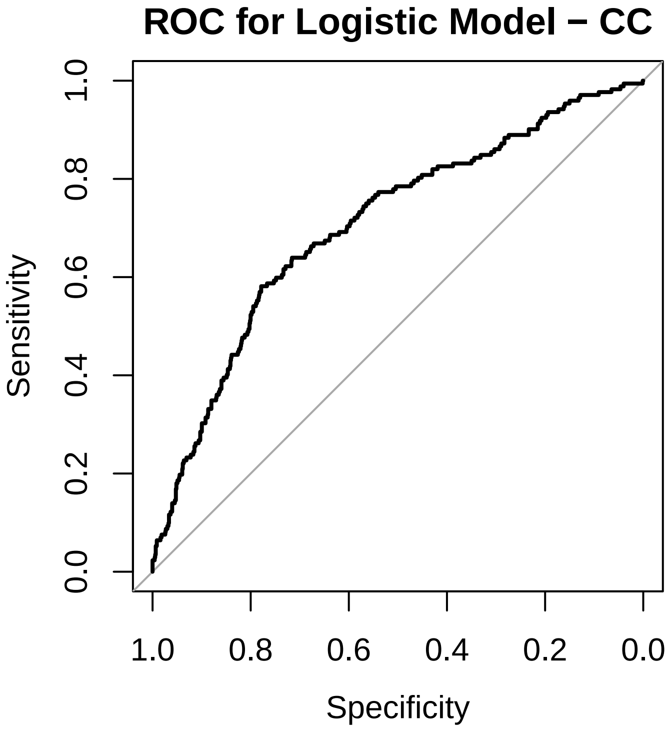
For less code, the 'MICE' package could have been used, referring to the function 'mice' for imputation of the training set and 'mice.reuse' to reuse the imputation models on the test set.

# Model

In the model section we will consider the two data sets; the complete case and imputed case. Both data sets are further divided into a train and test set. Since we want to do Lasso, we must standardize the data. The problem with this is data leakage. If we want to standardize the test data, we should standardize it using the mean and the standard deviation of the training data. Most importantly, using the test data to scale the test data will introduce correlation between the independent observations of the test set. Since the scaling information from the test set is "not available" to us at the time of training, we cannot expect the coefficients in the Lasso to be appropriately scaled compared to the test data. We solve this by scaling the training data, and then using the attributes of the training data to scale the test data accordingly.

Given the binary response it is natural to consider fitting a logistic regression model to our data. Although we intend to use lasso, it is nice to start by fitting a regular logistic regression model on the complete case data to get an indication of which covariates that are most present, and for later comparison. We obtain the regression coefficients, a confusion matrix, a ROC-curve and the ROC-AUC.

```
##                       Estimate Std. Error    z value  Pr(>|z|)
## (Intercept)         -8.1909517  0.8648439 -9.4710177 0.0000000
## male                 0.6102028  0.1348701  4.5243748 0.0000061
## age                  0.0644352  0.0082290  7.8302194 0.0000000
## educationhs         -0.3167564  0.1532992 -2.0662624 0.0388037
## educationcollege    -0.2142688  0.1813577 -1.1814710 0.2374157
## educationpost-grad -0.1121211  0.1954944 -0.5735261 0.5662885
## currentSmoker       -0.0122323  0.1911538 -0.0639921 0.9489765
## cigsPerDay           0.0209125  0.0076803  2.7228564 0.0064720
## BPMeds               0.3232638  0.2692250  1.2007202 0.2298598
## prevalentStroke     -0.0717723  0.7060112 -0.1016589 0.9190275
## prevalentHyp         0.3578473  0.1679542  2.1306245 0.0331201
## diabetes            -0.1037130  0.3737342 -0.2775047 0.7813926
## totChol              0.0025062  0.0013865  1.8075291 0.0706798
## sysBP                0.0155640  0.0046170  3.3710343 0.0007489
## diaBP               -0.0051849  0.0078536 -0.6601879 0.5091332
## BMI                 -0.0004882  0.0155704 -0.0313554 0.9749861
## heartRate           -0.0041223  0.0051729 -0.7968906 0.4255146
## glucose              0.0074510  0.0027470  2.7123748 0.0066803
##           Reference
## Prediction   0   1
##          0 917 161
##          1   9  11
```

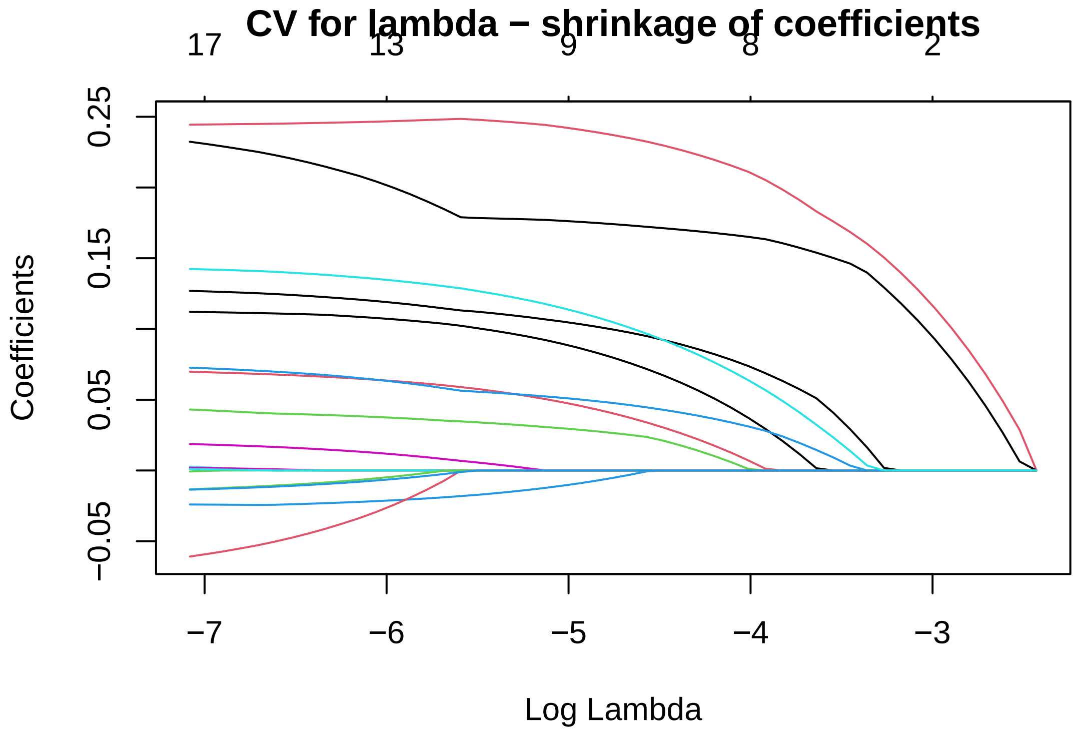


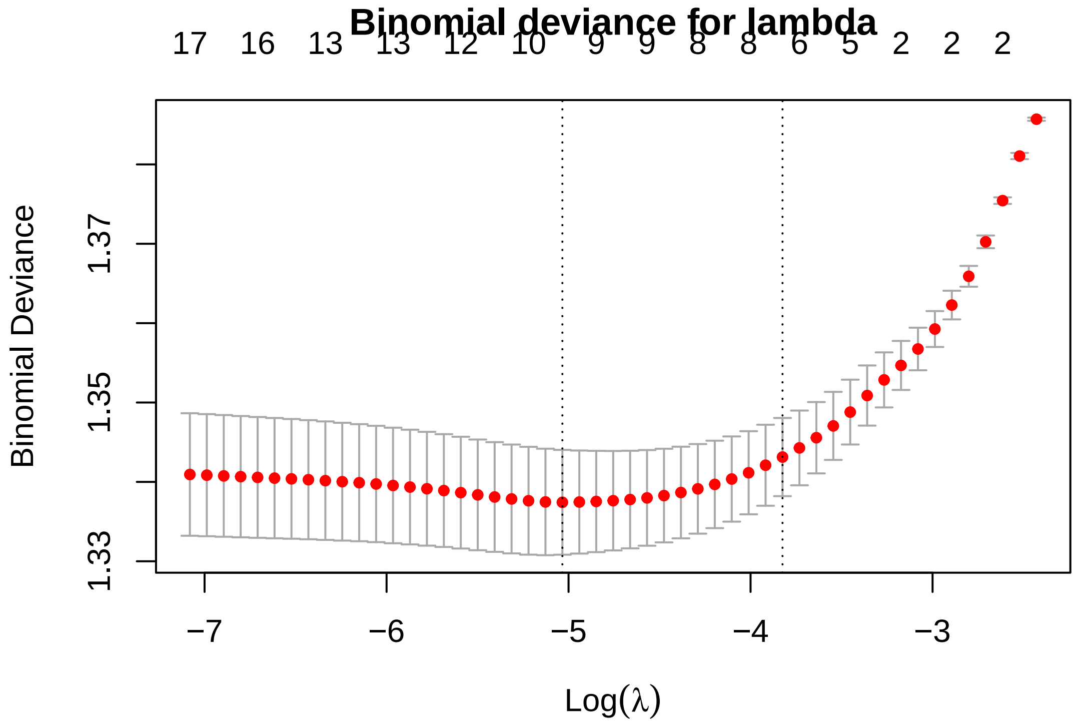
```
## Area under the curve: 0.7054
```

The logistic regression model chooses (Intercept), male, age, cigsPerDay, sysBP, glucose as the significant covariates, where the p-value cutoff is 0.01. It classifies very few positives correctly, which is very problematic if the model would be used to predict hearth disease.

### Lasso on Complete Case

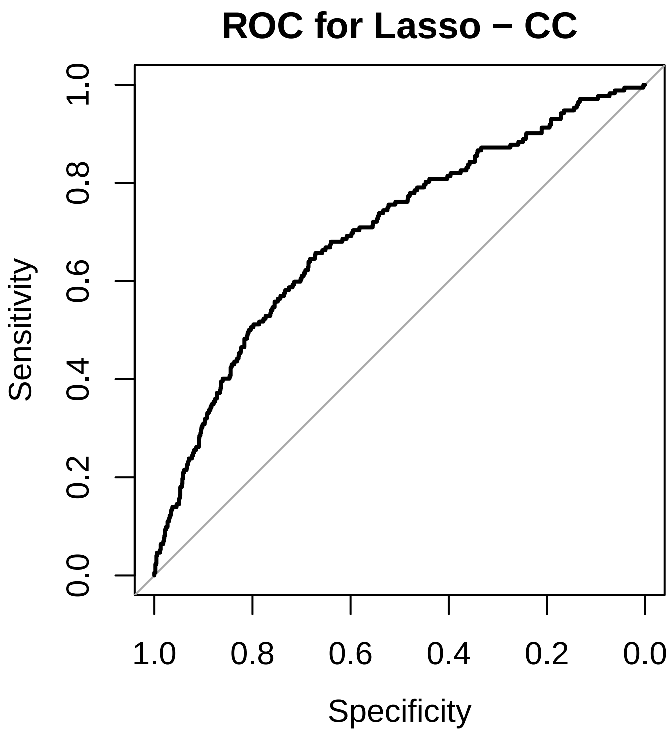
We continue to do the Lasso on the complete case data. To do this, we use cross-validation to find $\lambda_{min}$ and use the highest $\lambda$ with deviance within one standard deviation of $\lambda_{min}$. We cross-validate for $\lambda$ and plot the shrinkage and binomial deviance.


CV for lambda − shrinkage of coefficients
17
13
9
8
2
Coefficients
0.25
0.15
0.05
−0.05
−7
−6
−5
−4
−3
Log Lambda


Binomial deviance for lambda
17
16
13
13
12
10
9
9
8
8
6
5
2
2
2
Binomial Deviance
1.37
1.35
1.33
−7
−6
−5
−4
−3
Log(λ)

The confusion table, ROC and ROC-AUC is given below.

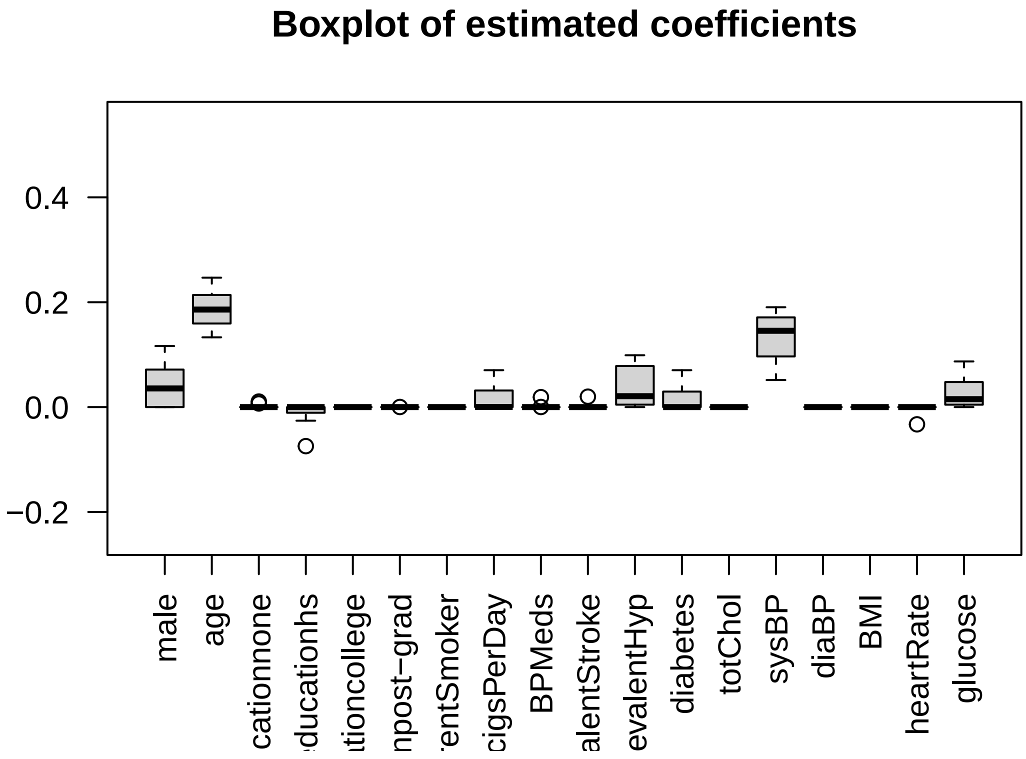
```
##           Reference
## Prediction   0   1
##          0 533  50
##          1 393 122
```

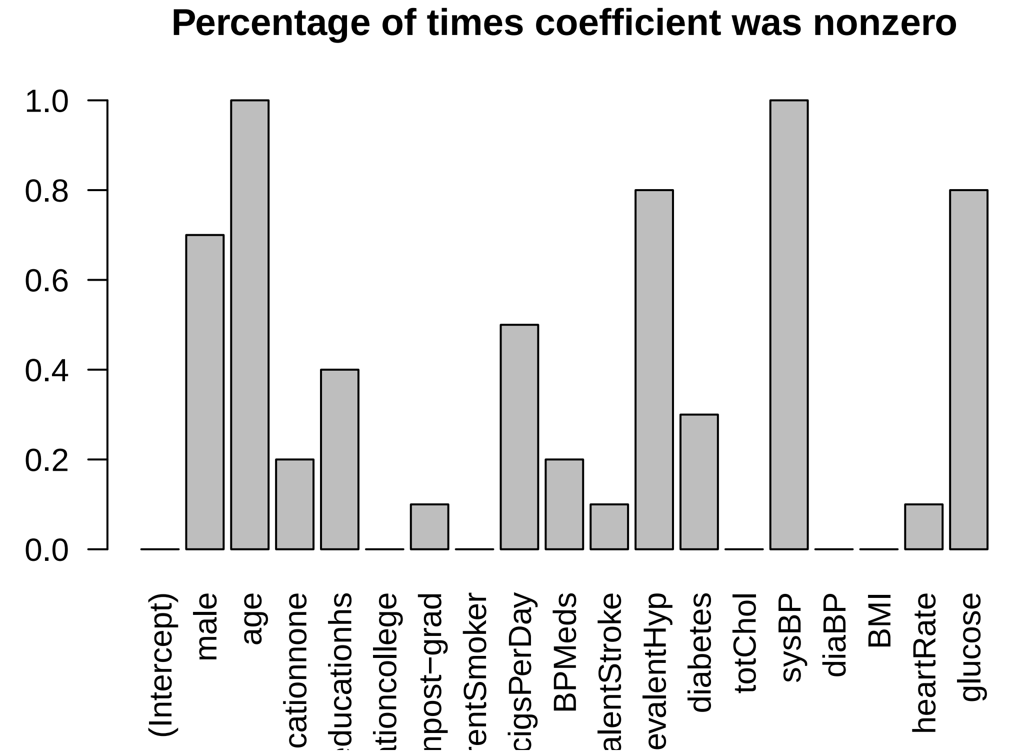


```
## Area under the curve: 0.6983
```

The Lasso on the complete case data chooses male, age, cigsPerDay, prevalentHyp, sysBP, glucose as the significant covariates. Initially, this is the same lasso model is much better at classifying positives than the full logistic model and has only slightly worse AUC, at 0.6982929 for the Lasso versus 0.705441 for the logistic model. We see that the Lasso includes 'prevalentHyp', while the logistic model finds it almost significant, with p-value 0.0331201. This agrees with the reasoning we made earlier, saying it could be an important parameter, but not as important as (either) 'sysBP' or 'diaBP'.

To obtain a better understanding of these coefficients, we bootstrap from the training data to fit Lasso models and store their coefficients.





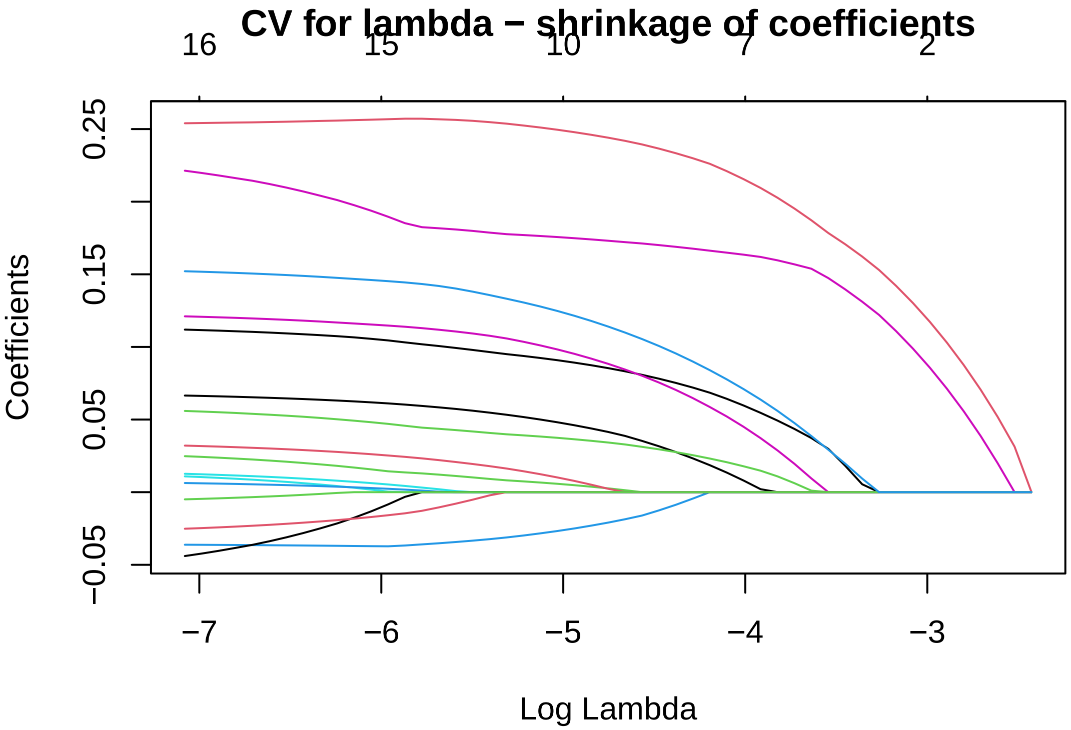
We ran the bootstrap using 10 iterations. The variables that have nonzero coefficients in the Lasso models at least 70% of the time, are male, age, sysBP, glucose. Similarly, by those to those who are included at least 50% of the time, we obtain male, age, cigsPerDay, prevalentHyp, sysBP, glucose. This is indeed similar to the ones we picked out earlier.

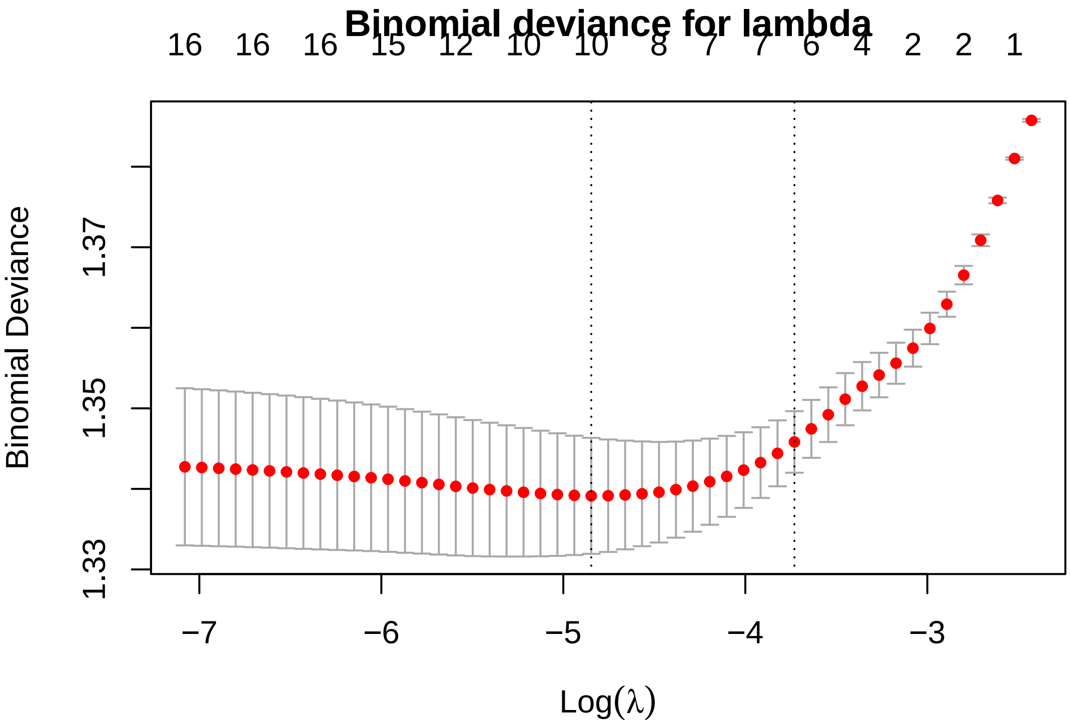
### Lasso on Imputed Data

We now do the same thing, just using the imputed data instead of the complete case. Even though the imputed data includes more samples, the data quality is going down when we impute. We expect similar results, but we must wait and see.

To get hands-on experience with the 'MICE' package, we also construct an imputed data set using 'mice' and 'mice.reuse', for comparison. 'mice' can be used on the entire training data, without needing to remove those samples with two or more covariates missing. The imputation model from the training set is emplyed to impute the test set as well, to avoid data leakage.

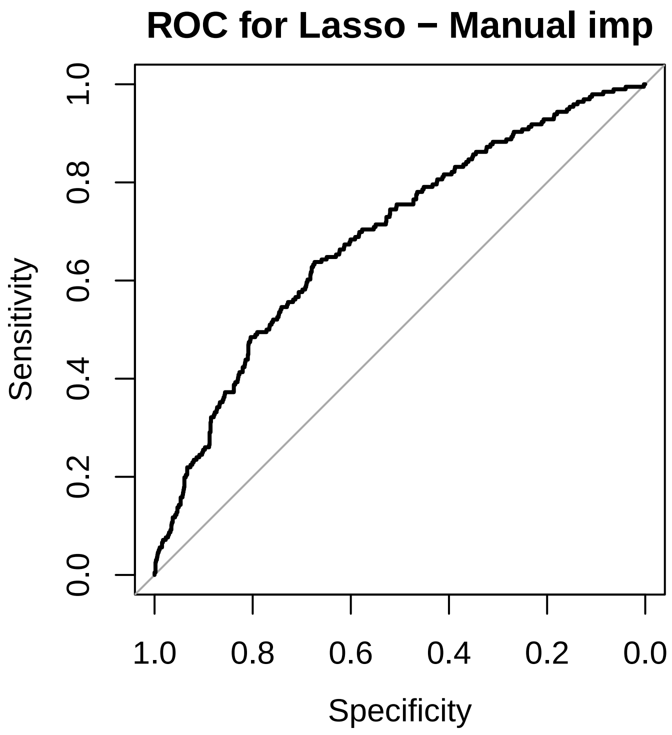
First we try the Lasso on the imputed dataset where we imputed with our manual technique. We show the shrinkage and the binomial deviance over $\lambda$, which is a part of the cross-validation





We also give the confusion matrix, the ROC and the ROC-AUC.

```
##            Reference
## Prediction   0   1
##          0 593  58
##          1 459 138
```



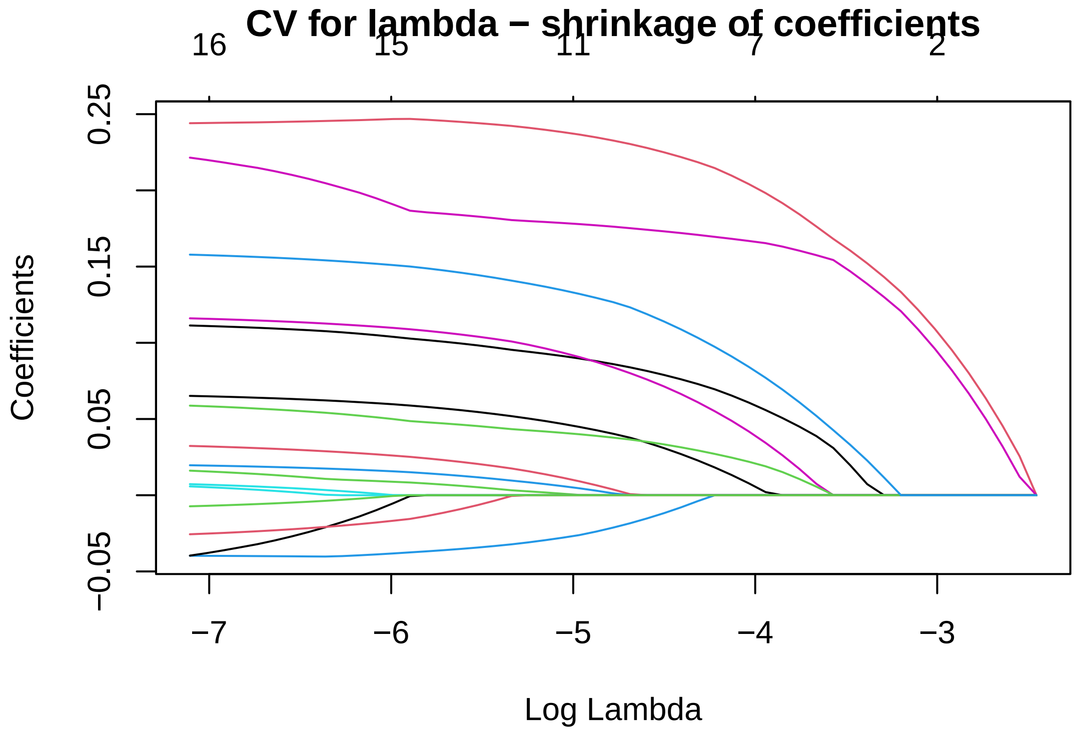
```
## Area under the curve: 0.6881
```

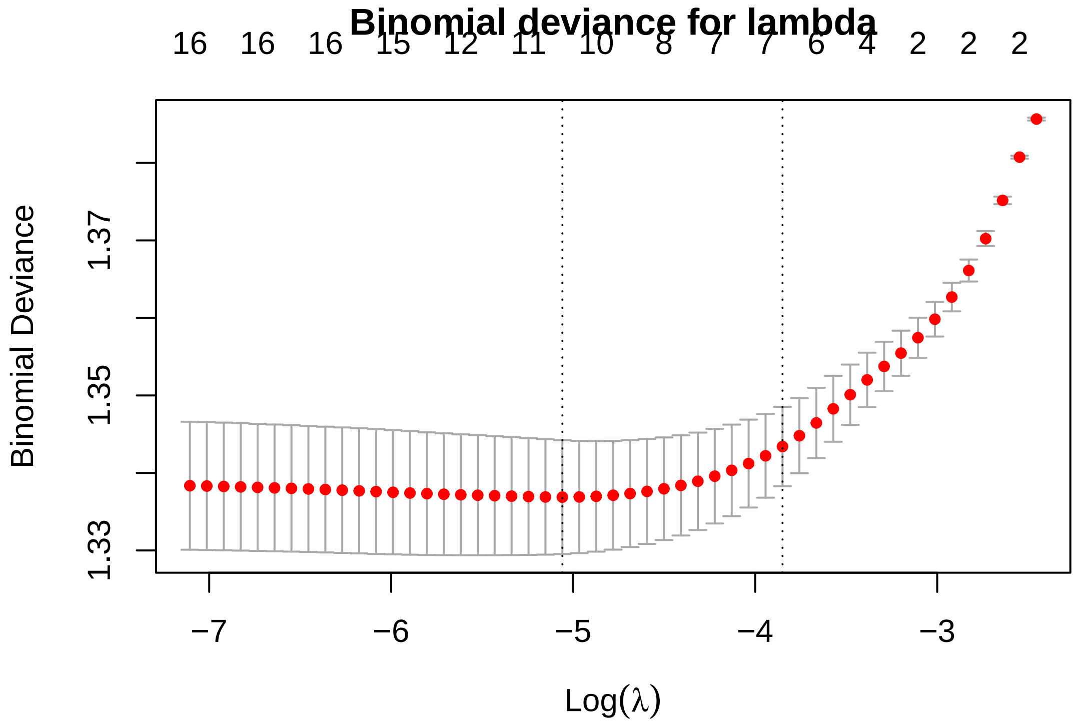
The performance of this model is quite similar to that of the complete case, which is to be expected. The Lasso on the manually imputed data chooses male, age, sysBP, glucose as the significant covariates.

Recall that the sensitivity can be measured by the true positive rate (i.e. the number of true positives over all positives) and the specificity can be measured by the true negative rate (i.e. true negative over all negative). Comparing the Lasso on the complete case data with the imputed data, we note that we have sensitivity 0.575594 in the complete case and sensitivity 0.5636882 in the imputed case. The specificity of the complete case Lasso is 0.7093023, while for the imputed case it is 0.7040816.

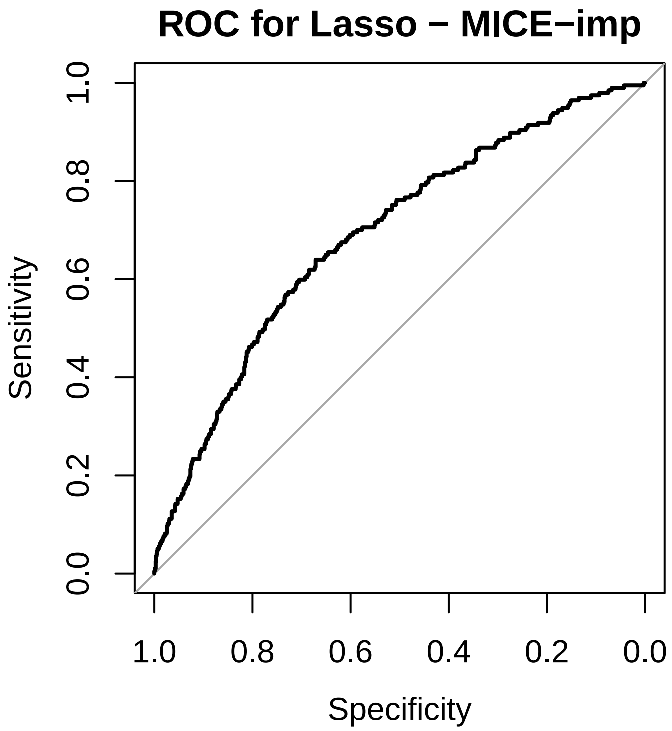
Although the difference is not huge, it may resemble that the data quality in the imputed data is slightly lower, although for selecting covariates, we obtained the same answer. In our data rich situation, this is neither clear enough to be rendered true, nor actually a problem, but for data poor situations, this is something to keep in mind.

We try to do the same thing, using the 'MICE'-imputed data set, first plotting the shrinkage and binomial deviance over $\lambda$, and then give the confusion matrix, ROC and ROC-AUC.


CV for lambda − shrinkage of coefficients
16
15
11
7
2
Coefficients
0.25
0.15
0.05
−0.05
−7
−6
−5
−4
−3
Log Lambda


Binomial deviance for lambda
16 16 16 15 12 11 10 8 7 7 6 4 2 2 2
Binomial Deviance
1.37
1.35
1.33
−7
−6
−5
−4
−3
Log(λ)

```
##           Reference
## Prediction   0   1
##          0 614  58
##          1 460 139
```



```
## Area under the curve: 0.6891
```

The predictive performance has not changed a lot, in the eyes of AUC.

The performance of this model is quite similar to that of the complete case, which is to be expected. The Lasso on the MICE-imputed data chooses male, age, cigsPerDay, prevalentHyp, sysBP, glucose as the significant covariates.

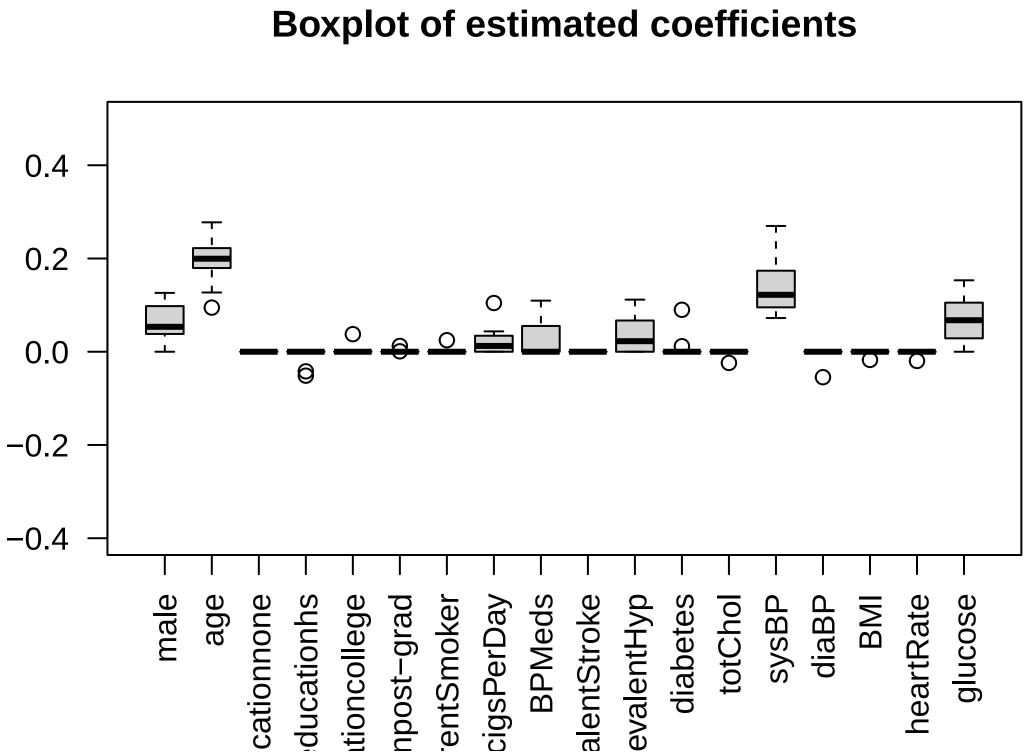
We suspect that the sensitivity and specificity is even lower for the MICE-imputed data, as it also imputes those with more than one missing value.

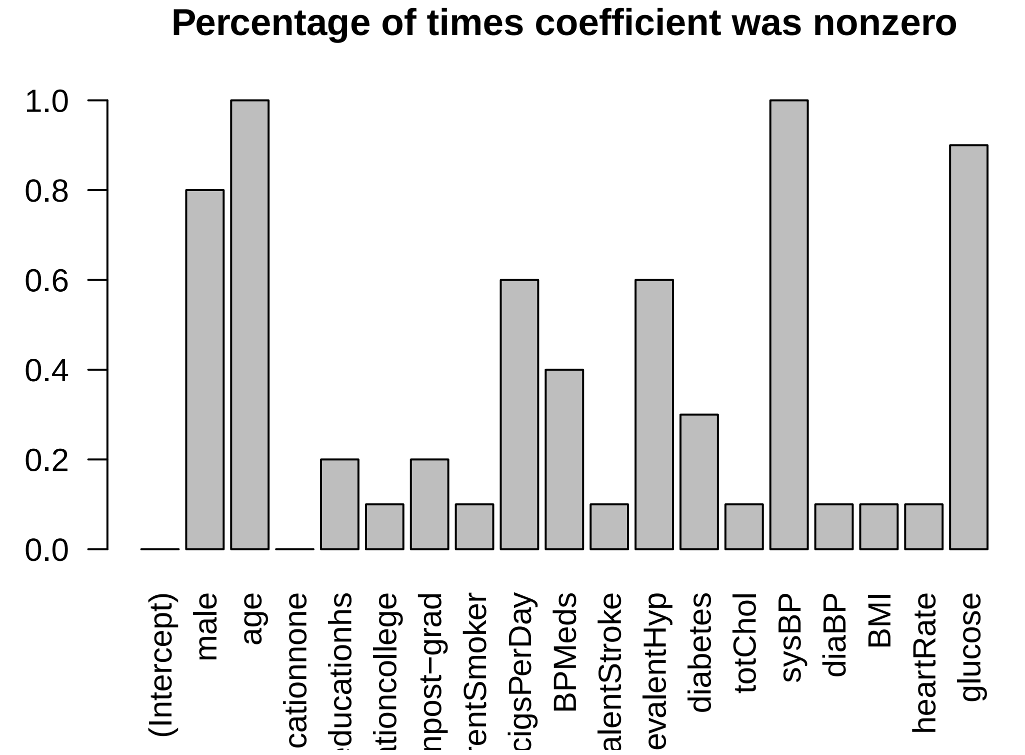
Comparing the Lasso on the complete case data with the imputed data, we note that we have sensitivity 0.5636882 in the manually imputed case and 0.5716946. The specificity of the manually imputed case Lasso is 0.7040816, while for the mice case it is 0.7055838.

Although the difference is not huge, it may look like our suspicions were wrong. The 'mice' function performs quite good imputation, but it is interesting that our manual imputation procedure manages almost as well in this case.

### Bootstrapping

To obtain improved estimates for the coefficients in the Lasso, we use bootstrapping. The data set used will be the MICE-imputed data set.





We ran the bootstrap using 10 iterations. The variables that have nonzero coefficients in the Lasso models at least 70% of the time, are male, age, sysBP, glucose. Similarly, by those to those who are included at least 50% of the time, we obtain male, age, cigsPerDay, prevalentHyp, sysBP, glucose. This is indeed similar to the ones we picked out earlier.

# Inference

In order to do inference we simply fit a logistic regression model using the `glm` function in `R`, and extract the inference from there. However, we will keep the coefficients chosen by the lasso-bootstrapping iterations in the earlier models, and now use the test data to fit a logistic model to avoid overfitting. Note that we have only used the test data to predict and observe different measures, such as ROC-AUC, sensitivity/specificity, and so on. The test data is therefore suitable for inference, as it has not been perturbed in the procedure of fitting the models.

We start out with the complete case models and fit a logistic model with the most important variables, namely male, age, sysBP, glucose. We first state the coefficients of the new regression model and their confidence intervals.

```
##                Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept) -8.42663685 0.727151909 -11.588551 4.710346e-31
## age          0.04891710 0.011168331   4.379982 1.186889e-05
## male         0.85267388 0.181851299   4.688852 2.747416e-06
## sysBP        0.01998341 0.003974812   5.027511 4.968880e-07
## glucose      0.01273926 0.003233364   3.939941 8.150151e-05
```

The confidence intervals of these variables for the new regression model (left) and naive logistic model fit (right) on the complete data set is given by the following.

```
##                2.5 %   97.5 %    2.5 %   97.5 %
## (Intercept) -9.89068 -7.03681 -9.90421 -6.51217
## age          0.02715  0.07099  0.04840  0.08067
## male         0.49970  1.21362  0.34667  0.87569
## sysBP        0.01226  0.02786  0.00651  0.02463
## glucose      0.00673  0.01950  0.00216  0.01295
```

We can observe that all the coefficients in the new model are significant! Comparing confidence intervals for the naive model and the new model with Lasso-selected variables, we see that the confidence intervals are shifted more towards zero, and some of them has even become slightly smaller. For example, the confidence interval of 'sysBP' went from 0.0181175 to 0.0156049 after the subset selection.

Similarly, we may fit a logistic model on the imputed data. We include the variables (male, age, sysBP, glucose) that was nonzero more than 70% of the times in the bootstrap, and we state the coefficients of the new regression model and their confidence intervals.

```
##                 Estimate  Std. Error    z value     Pr(>|z|)        2.5 %
## (Intercept) -7.485575148 0.687604857 -10.886449 1.337545e-27 -8.861047168
## age          0.056759300 0.010530334   5.390076 7.042786e-08  0.036267725
## male         0.386430057 0.172145588   2.244786 2.478185e-02  0.048874523
## sysBP        0.014070721 0.003614571   3.892777 9.910309e-05  0.006989930
## cigsPerDay   0.022224504 0.006700120   3.317031 9.097964e-04  0.009023368
## glucose      0.007009824 0.002804436   2.499548 1.243517e-02  0.001461963
##                   97.5 %
## (Intercept) -6.16225604
## age          0.07759168
## male         0.72451297
## sysBP        0.02118910
## cigsPerDay   0.03533988
## glucose      0.01264495
```

Again, we can observe that all the coefficients in the new model are significant!

We take a brief look at the complete case model with Lasso-selected variables to the model on the imputed data with Lasso-selected variables. The intervals are for the model on the imputed data (left) and the model on the complete data (right).

```
##             2.5 %    97.5 %    2.5 %     97.5 %
## (Intercept) "-8.861" "-6.1623" "-9.8907" "-7.0368"
## age         "0.0363" "0.0776"  "0.0272"  "0.071"
## male        "0.0489" "0.7245"  "0.4997"  "1.2136"
## sysBP       "0.007"  "0.0212"  "0.0123"  "0.0279"
## glucose     "0.0015" "0.0126"  "0.0067"  "0.0195"
## cigsPerDay  "0.009"  "0.0353"  "*"       "*"
```

We can see, for example by considering 'sysBP', see that the width of the interval has gone further down by working on the imputed data, as the confidence interval width for 'sysBP' was 0.7139159 for the complete case model and 0.6756384. This may simply be because we use more data to fit the model, but it may also be a nonsensical question, as we are in essence fitting two different models. The inclusion of 'cigsPerDay' in the model on imputed data is probably a key reason why we see such a difference. More or less, we obtain the same results, as it is the same covariates that come back time and time again.

# Discussion

When we set out, our goal was to explore imputation techniques and learn about 'MICE', single imputation, Lasso and bootstrapping with Lasso. We first fit a logistic regression model on the complete case training set and then a Lasso on the complete case data set. Bootstrapping was used to see how often each variable was included in a Lasso model. The variables that was present in at least 70% of the models, were male, age, sysBP, glucose. As discussed in the exploratory analysis, some variables are included that are highly correlated to others. For example, 'sysBP' has been included in every model we ran, and as mentioned, 'sysBP' is extremely correlated to 'diaBP', which probably forced 'diaBP' out of the model. If we select those variables that were in at least 50% of the models, we obtain male, age, cigsPerDay, prevalentHyp, sysBP, glucose, and we see that 'cigsPerDay' and 'prevalentHyp' is selected, as well. This is as expected, as these were some of the variables we mentioned could absorb some of the information from omitted variables, such as 'currentSmoker' and 'diaBP'.

The Lasso on imputed data chose male, age, sysBP, glucose, which matches the complete case quite well.

Another interesting thing we discovered, was that even though we obtain good results with the imputed data, the data quality seems to go down, if only barely. The sensitivity and specificity of the went down slightly when using the imputed data, but as the Lasso selects the same variables, it does not matter a lot for our purposes. The decrease in data quality should have been more visible if the percentage of imputed values were higher, or if we were in a not-so-data-rich setting.

What we can probably conclude, is that the variables male, age, sysBP, glucose are the most significant, and that we may include 'cigsPerDay' or even 'prevalentHyp' if we want to. Lesson learned: Try to avoid becoming an old, smoking man with high blood pressure, and let 'mice' impute your data instead of doing it manually.