Assignment 4

Matt Carmosino

2023-03-03

Assignment 4: Logistic Regression

For this assignment I will be using a dataset from kaggle that contains data on heart failure.

https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

```
library(psychTools)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

heart <- read.csv("C:/Users/matth/OneDrive/Desktop/heart.csv")
heart <- na.omit(heart)
```

Dataset Information

- Age: age of the patient [years]
- Sex: sex of the patient [M: Male, F: Female]
- ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP:
- Non-Anginal Pain, ASY: Asymptomatic]
- RestingBP: resting blood pressure [mm Hg]
- Cholesterol: serum cholesterol [mm/dl]
- FastingBS: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
- RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
- MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
- ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
- Oldpeak: oldpeak = ST [Numeric value measured in depression] ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- HeartDisease: output class [1: heart disease, 0: Normal]

Logical Regression

Research question Based on different characteristics of patient data, can we classify whether a patient has heart failure or not.

Ho (null): The classification of a patients heart failure is not possible in relation to these variables

Ha (alternative): The classification of a patients heart failure is possible in relation to at least one these variables

Variables of interest Since there are a lot of variables in this dataset, I am going to select some independent variables of interest. * Age * Sex (M or F) * Cholesterol (mm/dl) * RestingBP (mm Hg) * MaxHR (bpm)

And my independent variable will be HeartDisease. 1 being heart disease and 0 being normal

```
heart_interest <- heart %>% select(Age, Sex, Cholesterol, RestingBP, MaxHR, HeartDisease)
ls(heart_interest)
```

```
## [1] "Age"          "Cholesterol"  "HeartDisease" "MaxHR"        "RestingBP"
## [6] "Sex"
```

```
table(heart_interest$HeartDisease)
```

```
##
##   0   1
## 410 508
```

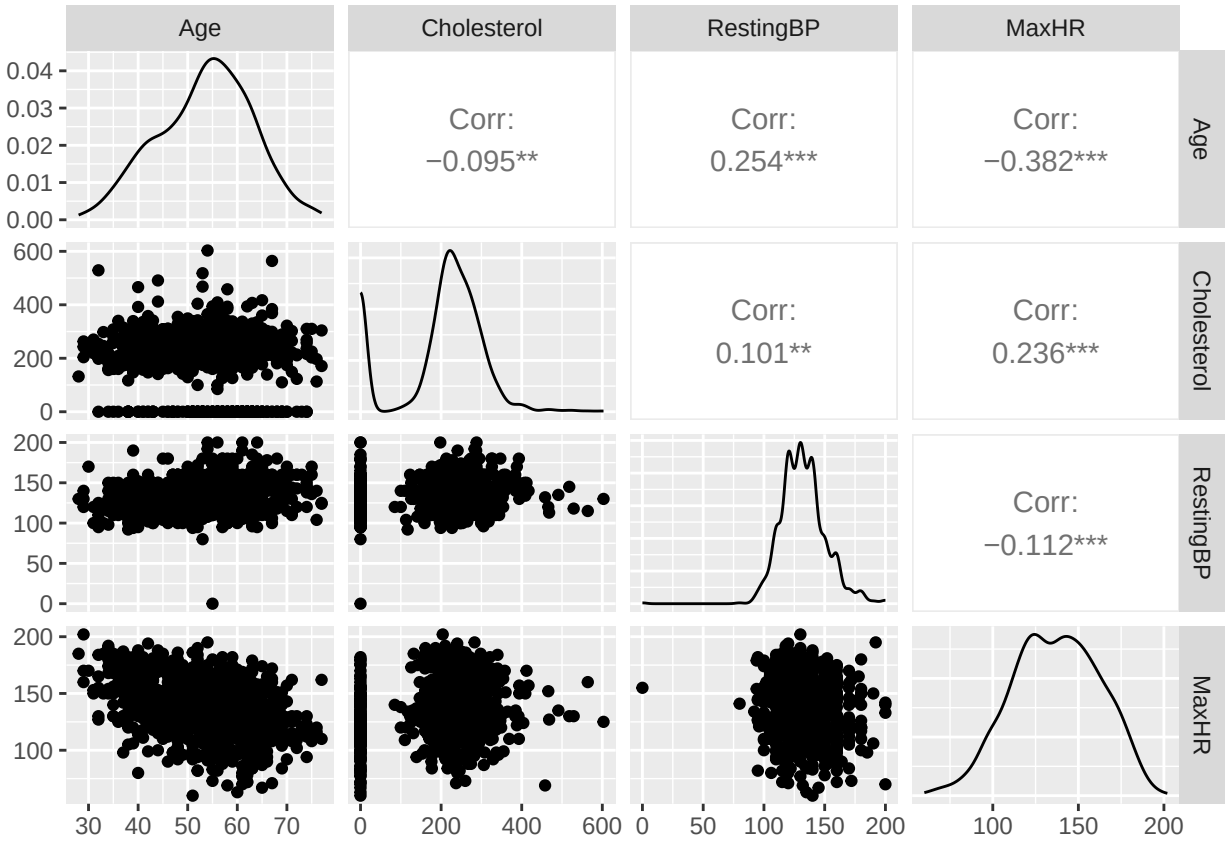
Now I need to wrangle Data. Since the HeartDisease variable is already binary, I do not need to convert to a factor, but for Sex I need to transform.

```
# Set levels where "F" = 1
heart_interest$Sex<- factor(
heart_interest$Sex,
levels = c("M", "F")
)
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
heart_interest %>%
  select(-Sex,-HeartDisease) %>% # remove categorical
  ggpairs()
```



Not much correlation between predictors, except for MaxHR and Age with 0.382

Now we normalize data

```
# Load {bestNormalize}
library(bestNormalize)

# Set seed
set.seed(1234)

# We don't want Sex or HeartDisease to be normalized since
# it is categorical
heart_numeric <- heart_interest %>% select(-Sex, -HeartDisease)

# Store in a list
normalized_list <- lapply(
  1:ncol(heart_numeric), # loop over columns
  function(column){

    # Apply and return best normalize
      return(
        bestNormalize(heart_numeric[,column])
      )

  }
)
```

```r
# Name the list
names(normalized_list) <- colnames(heart_numeric)

# Extract transformed values
transformed_list <- lapply(
  normalized_list,
  function(x){
    x$x.t # within each element
    # of our list, extract
    # the transformed values
  }
)

# Bring variables back together in a data frame
heart_normalized <- do.call(
  cbind.data.frame, transformed_list
)

# Initialize final dataset
heart_final <- heart_normalized

# Add back `Sex` and 'HeartDisease'
heart_final$Sex <- heart_interest$Sex
heart_final$HeartDisease <- heart_interest$HeartDisease
```

Perform Regression

```r
logm_heart <- glm(
  formula = HeartDisease ~ .,
  data = heart_final,
  family = "binomial"
)

summary(logm_heart)
```

```
##
## Call:
## glm(formula = HeartDisease ~ ., family = "binomial", data = heart_final)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2534  -0.9225   0.5018   0.8807   2.1790
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.55444    0.08514   6.512 7.43e-11 ***
## Age          0.35981    0.08705   4.133 3.57e-05 ***
## Cholesterol -0.10827    0.08412  -1.287    0.198
## RestingBP    0.11343    0.08094   1.401    0.161
## MaxHR       -0.74804    0.09181  -8.148 3.70e-16 ***
## SexF        -1.41743    0.19844  -7.143 9.14e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1262.1  on 917  degrees of freedom
## Residual deviance: 1019.7  on 912  degrees of freedom
## AIC: 1031.7
##
## Number of Fisher Scoring iterations: 4
```

Multicollinearity

```
car::vif(logm_heart)
```

```
##         Age Cholesterol   RestingBP       MaxHR         Sex
##    1.150061    1.072454    1.090350    1.091535    1.040627
```

None! Now to remove non-significant predictors

```
logm_heart <- glm(
  formula = HeartDisease ~ Age + MaxHR + Sex,
  data = heart_final,
  family = "binomial"
)

summary(logm_heart)
```

```
##
## Call:
## glm(formula = HeartDisease ~ Age + MaxHR + Sex, family = "binomial",
##     data = heart_final)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3736  -0.9272   0.5014   0.8827   2.1997
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.55632    0.08490   6.553 5.64e-11 ***
## Age          0.39001    0.08451   4.615 3.93e-06 ***
## MaxHR       -0.76246    0.09109  -8.370  < 2e-16 ***
## SexF        -1.45912    0.19514  -7.477 7.58e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1262.1  on 917  degrees of freedom
## Residual deviance: 1022.9  on 914  degrees of freedom
## AIC: 1030.9
##
## Number of Fisher Scoring iterations: 4
```

Now for odds-ratio

```
exp(coef(logm_heart))
```

```
## (Intercept)         Age       MaxHR        SexF
##   1.7442452   1.4769923   0.4665170   0.2324412
```

Interpretation: * For each one unit increase in Age the odds of Heart Disease increase by a factor of 1.477 holding other variables constant * For each one unit increase in MaxHR the odds of Heart Disease increase by a factor of 0.467 holding other variables constant * For SexF, someone that is a female (Sex=1) are 0.232 times lower of having heart disease compared to a male (Sex=0)

To get the odds ratio of males, just take the reciprocal of the SexF odds ratio 1/0.232 which gives us 4.31. Meaning that the odds of having heart disease for males, compared to females and holding all other variables constant, is 4.31 times higher. Snd since the p value is way less than 0.05, the gender difference in heart disease is statistically significant.

```
probs <- predict(
logm_heart,
type = "response"
# needed for probabilities
)
```

```
# Obtain classes

heart_final$HeartDisease <- factor(
  heart_final$HeartDisease,
levels = c(0, 1)
)

# I kept getting 'Error: `data` and `reference` should be factors with the same levels.'
# So I made heart disease a factor here even though it was already binary and it worked

classes <- factor(
ifelse(
probs > 0.50,
1, # if TRUE
0 # if FALSE
))
```

Evaluate Classification Confusion Matrix

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# Compute confusion matrix
confusionMatrix(
data = classes, # predicted
reference = heart_final$HeartDisease, # actual
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 251 103
##          1 159 405
##
##                Accuracy : 0.7146
##                  95% CI : (0.6842, 0.7436)
##     No Information Rate : 0.5534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4149
##
##  Mcnemar's Test P-Value : 0.000679
##
##             Sensitivity : 0.6122
##             Specificity : 0.7972
##          Pos Pred Value : 0.7090
##          Neg Pred Value : 0.7181
##              Prevalence : 0.4466
##          Detection Rate : 0.2734
##    Detection Prevalence : 0.3856
##       Balanced Accuracy : 0.7047
##
##        'Positive' Class : 0
##
```

```
# removed positive because i had trouble including it
```

Our model classifies heart diseases with 71.5% accuracy, not bad

Rms

```
# Load {rms}
library(rms)
```

```
## Loading required package: Hmisc
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     heart

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve
```

```r
# Fit model with `lrm`
lrm_heart <- lrm(
  formula = HeartDisease ~ Age + Sex + MaxHR,
  data = heart_final)

# Print summary
lrm_heart
```

```
## Logistic Regression Model
##
## lrm(formula = HeartDisease ~ Age + Sex + MaxHR, data = heart_final)
##
##                       Model Likelihood     Discrimination    Rank Discrim.
##                             Ratio Test            Indexes          Indexes
## Obs           918    LR chi2     239.27      R2       0.307    C       0.782
##  0            410    d.f.             3      R2(3,918)0.227    Dxy     0.565
##  1            508    Pr(> chi2) <0.0001    R2(3,680.7)0.293    gamma   0.565
## max |deriv| 3e-14                           Brier    0.188    tau-a   0.280
##
##           Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept  0.5563 0.0849  6.55  <0.0001
## Age        0.3900 0.0845  4.61  <0.0001
## Sex=F     -1.4591 0.1951 -7.48  <0.0001
## MaxHR     -0.7625 0.0911 -8.37  <0.0001
```

R-squared of 0.307, not the highest. Only 30% of the variance in the HeartDisease can be explained by our model. The C/AUC value of our model is 0.782 which means we have a good, almost strong model

```
table(classes)
```

```
## classes
##   0   1
## 354 564
```

```
table(heart_final$HeartDisease)
```

```
##
##   0   1
## 410 508
```

These tables show the misclassifications that occurred with the model.