MSDS 534 Statistical Learning Project (11/11/2020)

Christopher Rabeony

11/16/2020

## Data Attributes

* age - age in years.
* sex - gender (0 = male; 1 = female)
* cp - chest pain type
  + 1 = typical agina
  + 2 = atyptical angina
  + 3 = non-anginal pain
  + 4 = asymptomatic
* trestbps - resting blood pressure (in mm HG)
* chol - serum cholestoral in mg/dl
* fbs - fasting blood sugar > 120 mg/dl (0 = false; 1 = true)
* restecg - resting electrocardiographic results
  + 0 = normal
  + 1 = having ST-T Wave abnormal (T wave inversions and/or ST elevation or depression of > 005 mV)
  + 2 = showing probable or definite left vetricular hyptertropy
* thalach - maximum heart rate achieved in beats per minute (bpm)
* exang - exercise induced angina (0 = no; 1 = yes)
* oldpeak - ST depression induced by exercise relative to rest
* slope - the slope of the peak exercise ST segment
  + 1 - upsloping
  + 2 - flat
  + 3 - down-sloping
* ca - number of major vessels (0-3) colored by fluoroscopy
* thal - displays the thalassemia
  + 3 = normal
  + 6 = fixed defect
  + 7 = reversible defect
* goal - predicted target variable. Diagnosis of heart disease (angiographic disease status)
  + 0 = < 50% diameter narrowing
  + 1 = > 50% diameter narrowing)

# Set up libraries  
#library(pastecs)  
# Definition for clean tables  
#options(scipen=100)  
#options(digits=3)  
# Read the data into a data frame  
heart <- read.csv("E:/School/Statistical Learning for Data Science/Project/processed.cleveland (1).data", header = F)

#### Clean up our data.

names(heart) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "goal")#naming columns for convenience  
heart$ca <- as.integer(heart$ca)# manipulating the data for analysis  
heart$thal <- as.integer(heart$thal)  
heart$goal[heart$goal == 2] <- 1  
heart$goal[heart$goal == 3] <- 1  
heart$goal[heart$goal == 4] <- 1  
summary(heart)# checking mean, meadian, and number of NAs in table

## age sex cp trestbps   
## Min. :29.00 Min. :0.0000 Min. :1.000 Min. : 94.0   
## 1st Qu.:48.00 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:120.0   
## Median :56.00 Median :1.0000 Median :3.000 Median :130.0   
## Mean :54.44 Mean :0.6799 Mean :3.158 Mean :131.7   
## 3rd Qu.:61.00 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:140.0   
## Max. :77.00 Max. :1.0000 Max. :4.000 Max. :200.0   
##   
## chol fbs restecg thalach   
## Min. :126.0 Min. :0.0000 Min. :0.0000 Min. : 71.0   
## 1st Qu.:211.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:133.5   
## Median :241.0 Median :0.0000 Median :1.0000 Median :153.0   
## Mean :246.7 Mean :0.1485 Mean :0.9901 Mean :149.6   
## 3rd Qu.:275.0 3rd Qu.:0.0000 3rd Qu.:2.0000 3rd Qu.:166.0   
## Max. :564.0 Max. :1.0000 Max. :2.0000 Max. :202.0   
##   
## exang oldpeak slope ca   
## Min. :0.0000 Min. :0.00 Min. :1.000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.00 1st Qu.:1.000 1st Qu.:0.0000   
## Median :0.0000 Median :0.80 Median :2.000 Median :0.0000   
## Mean :0.3267 Mean :1.04 Mean :1.601 Mean :0.6722   
## 3rd Qu.:1.0000 3rd Qu.:1.60 3rd Qu.:2.000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :6.20 Max. :3.000 Max. :3.0000   
## NA's :4   
## thal goal   
## Min. :3.000 Min. :0.0000   
## 1st Qu.:3.000 1st Qu.:0.0000   
## Median :3.000 Median :0.0000   
## Mean :4.734 Mean :0.4587   
## 3rd Qu.:7.000 3rd Qu.:1.0000   
## Max. :7.000 Max. :1.0000   
## NA's :2

sapply(heart, sd, na.rm = TRUE)# checking standard deviations in table

## age sex cp trestbps chol fbs restecg   
## 9.0386624 0.4672988 0.9601256 17.5997477 51.7769175 0.3561979 0.9949713   
## thalach exang oldpeak slope ca thal goal   
## 22.8750033 0.4697945 1.1610750 0.6162261 0.9374383 1.9397058 0.4991195

heart\_matrix <- matrix(unlist(na.omit(heart)), nrow = 297, ncol= 14)# setting up for mass independence testing  
colnames(heart\_matrix) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "goal")  
ind<-combn(NCOL(heart\_matrix),2) # setting up combinations for testing indepence  
  
heartchi <- apply(ind, 2, function(ind) chisq.test(heart\_matrix[, ind[1]], heart\_matrix[, ind[2]]))# perfoming chi square indpendence tests  
heartchip <- data.frame(heartchi[[1]]$p.value)# extracting pvalue to start data fram  
heartchip

## heartchi..1...p.value  
## 1 0.4465554

for(i in 2:NROW(heartchi))# filling out table using loop  
{  
 heartchip <- cbind(heartchip, heartchi[[i]]$p.value)  
}  
heartchip <- t(heartchip)# transposing loop for formatting  
  
heartchi2<-cbind(t(ind), heartchi)# combining relevant tables for final display  
heartchi2<-cbind(heartchi2, heartchip[,1])  
heartchi2 #final product

## heartchi   
## heartchi..1...p.value 1 2 List,9 0.4465554   
## heartchi[[i]]$p.value 1 3 List,9 0.004393272   
## heartchi[[i]]$p.value 1 4 List,9 0.9996873   
## heartchi[[i]]$p.value 1 5 List,9 0.997342   
## heartchi[[i]]$p.value 1 6 List,9 0.5482317   
## heartchi[[i]]$p.value 1 7 List,9 1.075682e-06  
## heartchi[[i]]$p.value 1 8 List,9 0.0002833719  
## heartchi[[i]]$p.value 1 9 List,9 0.09549829   
## heartchi[[i]]$p.value 1 10 List,9 7.25608e-13   
## heartchi[[i]]$p.value 1 11 List,9 0.591535   
## heartchi[[i]]$p.value 1 12 List,9 0.02127395   
## heartchi[[i]]$p.value 1 13 List,9 0.7952163   
## heartchi[[i]]$p.value 1 14 List,9 0.07936753   
## heartchi[[i]]$p.value 2 3 List,9 0.08090545   
## heartchi[[i]]$p.value 2 4 List,9 0.6832768   
## heartchi[[i]]$p.value 2 5 List,9 0.2779731   
## heartchi[[i]]$p.value 2 6 List,9 0.621838   
## heartchi[[i]]$p.value 2 7 List,9 0.1558355   
## heartchi[[i]]$p.value 2 8 List,9 0.05002575   
## heartchi[[i]]$p.value 2 9 List,9 0.0191724   
## heartchi[[i]]$p.value 2 10 List,9 0.7495941   
## heartchi[[i]]$p.value 2 11 List,9 0.6870509   
## heartchi[[i]]$p.value 2 12 List,9 0.1344108   
## heartchi[[i]]$p.value 2 13 List,9 1.167601e-10  
## heartchi[[i]]$p.value 2 14 List,9 2.94569e-06   
## heartchi[[i]]$p.value 3 4 List,9 0.5377796   
## heartchi[[i]]$p.value 3 5 List,9 0.9524274   
## heartchi[[i]]$p.value 3 6 List,9 0.1510645   
## heartchi[[i]]$p.value 3 7 List,9 0.1711171   
## heartchi[[i]]$p.value 3 8 List,9 0.09051787   
## heartchi[[i]]$p.value 3 9 List,9 2.152552e-13  
## heartchi[[i]]$p.value 3 10 List,9 0.04369085   
## heartchi[[i]]$p.value 3 11 List,9 0.0002085757  
## heartchi[[i]]$p.value 3 12 List,9 0.0003266445  
## heartchi[[i]]$p.value 3 13 List,9 5.727237e-07  
## heartchi[[i]]$p.value 3 14 List,9 1.178284e-16  
## heartchi[[i]]$p.value 4 5 List,9 0.998125   
## heartchi[[i]]$p.value 4 6 List,9 0.1014211   
## heartchi[[i]]$p.value 4 7 List,9 0.000365159   
## heartchi[[i]]$p.value 4 8 List,9 0.0001014522  
## heartchi[[i]]$p.value 4 9 List,9 0.2392097   
## heartchi[[i]]$p.value 4 10 List,9 0.5649726   
## heartchi[[i]]$p.value 4 11 List,9 0.5898217   
## heartchi[[i]]$p.value 4 12 List,9 0.04103016   
## heartchi[[i]]$p.value 4 13 List,9 0.03851764   
## heartchi[[i]]$p.value 4 14 List,9 0.5343992   
## heartchi[[i]]$p.value 5 6 List,9 0.9134899   
## heartchi[[i]]$p.value 5 7 List,9 0.2237662   
## heartchi[[i]]$p.value 5 8 List,9 0.1316906   
## heartchi[[i]]$p.value 5 9 List,9 0.6462736   
## heartchi[[i]]$p.value 5 10 List,9 0.0004816346  
## heartchi[[i]]$p.value 5 11 List,9 0.5782789   
## heartchi[[i]]$p.value 5 12 List,9 0.5666649   
## heartchi[[i]]$p.value 5 13 List,9 0.06036069   
## heartchi[[i]]$p.value 5 14 List,9 0.09511112   
## heartchi[[i]]$p.value 6 7 List,9 0.3509379   
## heartchi[[i]]$p.value 6 8 List,9 0.4417411   
## heartchi[[i]]$p.value 6 9 List,9 1   
## heartchi[[i]]$p.value 6 10 List,9 0.8129077   
## heartchi[[i]]$p.value 6 11 List,9 0.1463607   
## heartchi[[i]]$p.value 6 12 List,9 0.04835467   
## heartchi[[i]]$p.value 6 13 List,9 0.1836237   
## heartchi[[i]]$p.value 6 14 List,9 1   
## heartchi[[i]]$p.value 7 8 List,9 0.01077514   
## heartchi[[i]]$p.value 7 9 List,9 0.2802171   
## heartchi[[i]]$p.value 7 10 List,9 4.695767e-10  
## heartchi[[i]]$p.value 7 11 List,9 0.03494283   
## heartchi[[i]]$p.value 7 12 List,9 0.2417338   
## heartchi[[i]]$p.value 7 13 List,9 0.5219841   
## heartchi[[i]]$p.value 7 14 List,9 0.008331151   
## heartchi[[i]]$p.value 8 9 List,9 0.02772094   
## heartchi[[i]]$p.value 8 10 List,9 5.114123e-07  
## heartchi[[i]]$p.value 8 11 List,9 0.01408949   
## heartchi[[i]]$p.value 8 12 List,9 0.5982188   
## heartchi[[i]]$p.value 8 13 List,9 0.006604571   
## heartchi[[i]]$p.value 8 14 List,9 0.07182862   
## heartchi[[i]]$p.value 9 10 List,9 0.01838549   
## heartchi[[i]]$p.value 9 11 List,9 9.243841e-06  
## heartchi[[i]]$p.value 9 12 List,9 0.004750047   
## heartchi[[i]]$p.value 9 13 List,9 1.262202e-07  
## heartchi[[i]]$p.value 9 14 List,9 9.510884e-13  
## heartchi[[i]]$p.value 10 11 List,9 4.03303e-16   
## heartchi[[i]]$p.value 10 12 List,9 0.0003495227  
## heartchi[[i]]$p.value 10 13 List,9 5.136779e-07  
## heartchi[[i]]$p.value 10 14 List,9 3.151884e-05  
## heartchi[[i]]$p.value 11 12 List,9 0.0822788   
## heartchi[[i]]$p.value 11 13 List,9 1.160399e-06  
## heartchi[[i]]$p.value 11 14 List,9 3.630107e-10  
## heartchi[[i]]$p.value 12 13 List,9 0.001148854   
## heartchi[[i]]$p.value 12 14 List,9 1.372578e-15  
## heartchi[[i]]$p.value 13 14 List,9 1.241673e-18

lmheart <- glm(goal~., heart, family = "binomial")# creating basic, preliminary logistic model  
summary(lmheart)

##   
## Call:  
## glm(formula = goal ~ ., family = "binomial", data = heart)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7818 -0.5207 -0.1863 0.4248 2.3622   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.372042 2.879476 -2.560 0.01046 \*   
## age -0.014164 0.023970 -0.591 0.55459   
## sex 1.312073 0.488474 2.686 0.00723 \*\*   
## cp 0.575898 0.191197 3.012 0.00259 \*\*   
## trestbps 0.024044 0.010730 2.241 0.02504 \*   
## chol 0.004995 0.003774 1.324 0.18561   
## fbs -1.021918 0.555330 -1.840 0.06574 .   
## restecg 0.245153 0.185005 1.325 0.18513   
## thalach -0.020665 0.010225 -2.021 0.04327 \*   
## exang 0.926104 0.413343 2.241 0.02506 \*   
## oldpeak 0.247386 0.211832 1.168 0.24287   
## slope 0.570009 0.363085 1.570 0.11644   
## ca 1.267719 0.265384 4.777 1.78e-06 \*\*\*  
## thal 0.343936 0.100361 3.427 0.00061 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 409.95 on 296 degrees of freedom  
## Residual deviance: 204.69 on 283 degrees of freedom  
## (6 observations deleted due to missingness)  
## AIC: 232.69  
##   
## Number of Fisher Scoring iterations: 6