Logistic Regression/Classification

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Objectives

- What are possible risk factors related to heart diseases? What determines an employee being a desirable one for the firm? How to tell wehther a review in Amazon is real or not? How to model a categorical variable conveniently and efficiently?
- Logistic regression models are the most commonly used methods to model the probability of an event.
- We then automatically get a linear classification rules.
- Various criteria are introduced.
- Analogous to least squared solutions for the usual regression models, we use maximum likelihood estimations.

Objectives

0. Introduction
1. A quick EDA
2. Logistic Regressions (Illustrated with one predictor)

- Link Function
- Maximum Likelihood Estimator (MLE)
  - Min cross entropy
- Inference for the Coefficients
  - Wald Intervals / Tests (through the MLE's)

Objectives

3. Classification

- Classification rules: thresholding $p(y = 1|x)$
- Criteria
  - Misclassification errors

4. Multiple Logistic Regressions

- Natural Extension
- Model selection through backward selection
- Final model
- Classification

Objectives

5. Training/Testing/Validation data
6. R functions

  - `glm()/Anova()`

Read

- Chapter 4.1-4.3.4
- Chapter 20.1-20.5

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
    - Classification rules
    - Criterion for classifier
6. Multiple Logistic Regression and Classification
    - Multiple logistic regression
    - Final model
7. Classification Revisit
    - Two features
    - Final model
8. Training/Testing Data

Case Background

- Heart disease is the leading cause of the death in United States. One out of four deaths is due to heart disease.
- It is important to identify Coronary Heart Disease risk factors.
- Many studies have indicate that, high blood pressure, high cholesterol, age, gender, race are among some major risk factors.
- Starting from the late 40th, National Heart, Lung and Blood Institute (NHLBI) launched it's famous Framingham Heart Study. By now subjects of three generations together with other people have been monitored and followed in the study.
- Over thousands research papers have been published using these longitudinal data sets. More details.
- Using a piece of the data gathered at the beginning of the study, we illustrate how to identify risk factors of heart disease and how to classify one with a heart disease.
- More details: https://www.framinghamheartstudy.org/fhs-about/

Data Overview

1,406 participants. Conditions gathered at the beginning of the study (early 50s).

Variable	Description
`Heart Disease`	Indicator of having heart disease or not
`AGE`	Age
`SEX`	Gender
`SBP`	Systolic blood pressure
`DBP`	Diastolic blood pressure
`CHOL`	Cholesterol level
`FRW`	age and gender adjusted weight
`CIG`	Self-reported number of cigarettes smoked each week

Goal of the Study

- Identify risk factors
- Predict the probability of one with `Heart Disease`
- Predict if one has `Heart Desease` given the information in hand
- In particular,

Predict $Prob(HD = 1)$ for Alice, who is:

AGE	SEX	SBP	DBP	CHOL	FRW	CIG
45	FEMALE	100	80	180	110	5

where

$$HD = \begin{cases} 1 & \text{if heart disease} \\ 0 & \text{if normal} \end{cases}$$

EDA

Read Framingham.dat as fram_data

```
fram_data <- read.csv("data/Framingham.dat", sep=",", header=T, as.is=T)
str(fram_data)
```

```
## ’data.frame’:    1407 obs. of  8 variables:
##  $ Heart.Disease.: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AGE           : int  45 49 47 48 49 47 48 48 46 45 ...
##  $ SEX           : chr  "MALE" "MALE" "MALE" "MALE" ...
##  $ SBP           : int  90 100 100 108 108 108 108 110 110 110 ...
##  $ DBP           : int  50 64 70 70 75 68 70 70 80 72 ...
##  $ CHOL          : int  216 237 215 340 149 165 196 229 204 183 ...
##  $ FRW           : int  76 97 86 93 95 88 79 85 112 93 ...
##  $ CIG           : int  5 0 50 0 0 0 20 25 0 20 ...
```

```
names(fram_data)
```

```
## [1] "Heart.Disease." "AGE"            "SEX"            "SBP"
## [5] "DBP"            "CHOL"           "FRW"            "CIG"
```

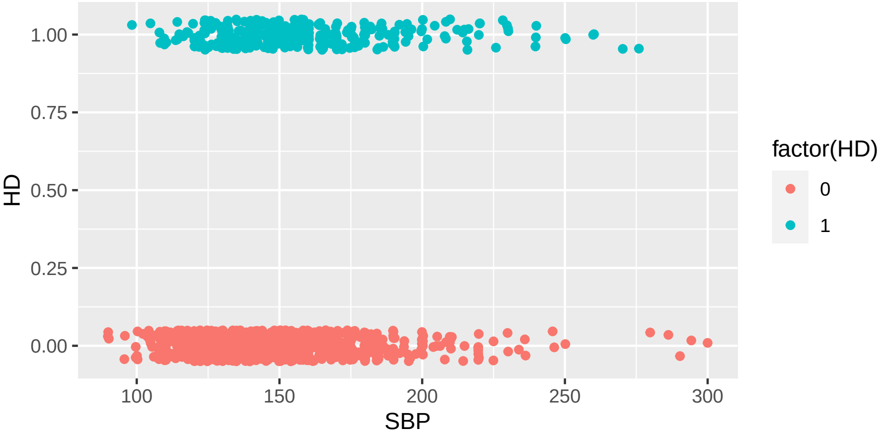
```
# dplyr way
fram_data %<>%
   rename(HD = Heart.Disease.) %>%
   mutate(HD = as.factor(HD),
          SEX = as.factor(SEX))

fram_data.new <- fram_data[1407,]
fram_data <- fram_data[-1407,]
```

HD vs SBP

The plot does not show the proportion of “1”’s vs. “0”’s as a function of SBP.

```
fram_data %>% mutate(HD = as.numeric(HD)-1) %>%
  ggplot(aes(x=SBP, y=HD)) +
  geom_jitter(height = .05, aes(color = factor(HD)))
```



HD vs SBP

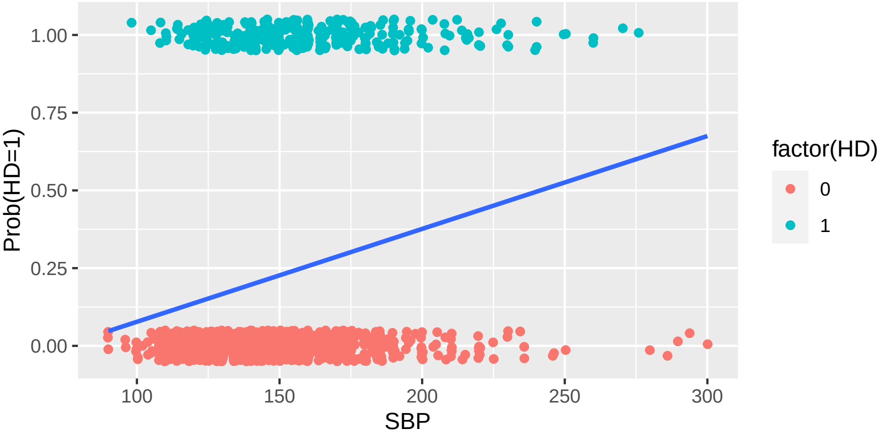
Can we use a linear model?

- For a binary response with a $0/1$ coding as `HD`, it can be shown that $X\hat{\beta}$ is in fact an estimate of $P(HD = 1|X)$ and we can predict the heart disease, say, if $\hat{P}(HD = 1|X) > 0.5$ and normal otherwise.
- However, some of our estimates maybe outside of $[0, 1]$, which makes no sense to interpret as probabilities.

HD vs SBP

```
fram_data %>% mutate(HD = as.numeric(HD)-1) %>%
  ggplot(aes(x=SBP, y=HD)) +
  geom_jitter(height = .05, aes(color = factor(HD))) +
  geom_smooth(method = "lm", se = FALSE) +
  ylab("Prob(HD=1)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
    - Classification rules
    - Criterion for classifier
6. Multiple Logistic Regression and Classification
    - Multiple logistic regression
    - Final model
7. Classification Revisit
    - Two features
    - Final model
8. Training/Testing Data

Logistic Regression: HD~SBP

We clearly would like to model the probability of one with HD given SBP. One of the most popular models is the **logistic regression model**.

Logistic model

In a logistic regression model, we will model the probability of one being sick as follows:

$$P(HD = 1|SBP) = \frac{e^{\beta_0+\beta_1 SBP}}{1 + e^{\beta_0+\beta_1 SBP}}$$

where β_0 and β_1 are unknown parameters. We see the following properties immediately:

i. $$0 < P(HD = 1|SBP) = \frac{e^{\beta_0+\beta_1 SBP}}{1 + e^{\beta_0+\beta_1 SBP}} < 1$$

ii. We get the $P(HD = 0|SBP)$:

$$P(HD = 0|SBP) = 1 - P(HD = 1|SBP) = \frac{1}{1 + e^{\beta_0+\beta_1 SBP}}$$

Logistic model

iii. What does the linear function describe? It is `log odds` of being `HD`.

$$logit(P(HD=1|SBP)) = \log\left(\frac{P(HD=1|SBP)}{P(HD=0|SBP)}\right) = \beta_0 + \beta_1 \times SBP$$

iv. The interpretation of β_1 is the change in log odds for a unit change in `SBP`.

v. $P(HD=1|SBP)$ is a monotone function of `SBP`, depending on the sign of β_1. $P(HD=1|SBP)$ is an increasing function of `SBP` if $\beta_1 > 0$.

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Maximum Likelihood Estimators (MLE)

- For our setting, the response is a categorical variable. The notion of least squared errors does not apply here.
- We will need to find some sensible function to minimize or maximize to estimate the unknown parameters β's.
- We introduce Likelihood Function of β_0 and β_1 given the data, namely the **Probability of seeing the actual outcome in the data.**

A Glimpse of the Data

Take a look at a piece of the data, randomly chosen from our data set. We can then see part of the likelihood function.

```
set.seed(2)
fram_data[sample(1:1406, 10), c("HD", "SBP")]
```

```
     HD SBP
975   0 135
710   0 124
774   0 136
416   1 130
392   1 120
273   0 134
1373  0 154
1228  0 150
1321  1 210
690   0 118
```

Maximum Likelihood Estimators (MLE)

We use this to illustrate the notion of likelihood function.

$$\begin{aligned}
\mathcal{L}(\beta_0, \beta_1|\mathrm{D}ata) &= Prob(\text{the outcome of the data}) \\
&= Prob((HD = 0|SBP = 135), (HD = 0|SBP = 124), \ldots, (HD = 1|SBP = 130), \ldots) \\
&= Prob(HD = 0|SBP = 135) \times Prob(HD = 0|SBP = 124) \times \ldots \times Prob(HD = 1|SBP = 130) \times \ldots) \\
&= \frac{1}{1 + e^{\beta_0 + 130\beta_1}} \cdot \frac{1}{1 + e^{\beta_0 + 140\beta_1}} \cdots \frac{e^{\beta_0 + 130\beta_1}}{1 + e^{\beta_0 + 130\beta_1}} \cdots
\end{aligned}$$

MLE: The estimate $(\hat{\beta}_1, \hat{\beta}_0)$ that maximizes the likelihood function is termed as `Maximum Likelihood Estimators`:

$$(\hat{\beta}_1, \hat{\beta}_0) = \arg\max_{\beta_0, \beta_1} \mathcal{L}(\beta_0, \beta_1|Data)$$

Maximum Likelihood Estimators (MLE)

Remark:

- MLE: $\hat{\beta}_0$ and $\hat{\beta}_1$ are obtained through $\max \log(\mathcal{L}(\beta_0, \beta_1|D))$
- Cross entropy: MLE can be obtained equivalently by

$$\begin{aligned} &\min -\frac{1}{n}\{\log(\mathcal{L}(\beta_0, \beta_1|D))\} \\ =&\min -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) \end{aligned}$$

- MLE can only be obtained through numerical calculations.

glm() function

glm() will be used to do logistic regression. It is very similar to lm() but some output might be different.

The default is logit link in glm

```r
fit1 <- glm(HD~SBP, fram_data, family=binomial(logit))
#summary(glm(as.numeric(HD)~SBP, fram_data, family="gaussian"))
summary(fit1)
```

```
##
## Call:
## glm(formula = HD ~ SBP, family = binomial(logit), data = fram_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.669  -0.711  -0.625  -0.524   2.107
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.66342    0.34602   -10.6  < 2e-16 ***
## SBP          0.01590    0.00221     7.2  6.2e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1485.9  on 1405  degrees of freedom
## Residual deviance: 1432.8  on 1404  degrees of freedom
## AIC: 1437
##
## Number of Fisher Scoring iterations: 4
```

Prediction

To see the prob function estimated by glm:

- logit = -3.66 + 0.0159 SBP
- $$\hat{P}(HD = 1|SBP) = \frac{e^{-3.66+0.0159\times SBP}}{1 + e^{-3.66+0.0159\times SBP}}$$
  $$\hat{P}(HD = 0|SBP) = \frac{1}{1 + e^{-3.66+0.0159\times SBP}}$$

Prediction

Now to estimate $P(HD = 1)$ for Alice, we can plug in her SBP=100 into the logit function:

```
fram_data.new
```

```
##        HD AGE    SEX SBP DBP CHOL FRW CIG
## 1407 <NA>  45 FEMALE 100  80  180 110   5
```

Based on fit1 we plug in SBP value into the prob equation.

$$\hat{P}(HD = 1|SBP = 100) = \frac{e^{-3.66+0.0159\times SBP}}{1 + e^{-3.66+0.0159\times SBP}} = \frac{e^{-3.66+0.0159\times 100}}{1 + e^{-3.66+0.0159\times 100}} \approx 0.112$$

We can also use the `predict()` function.

```
fit1.predict <- predict(fit1, fram_data.new, type="response")
fit1.predict
```

```
##  1407
## 0.112
```

Interpretations

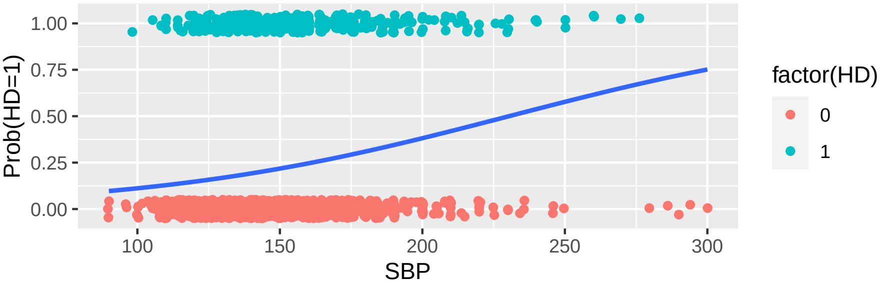
Let us see what we can say about the risk of HD when SBP increases.

- logit $=$ -3.66 $+$ 0.0159 SBP. That means log odds increases .0159 when SBP increases by 1.
- Notice the $Prob(HD = 1)$ is an increasing function of SBP since $\hat{\beta}_1 = .0159 > 0$. That means when SBP increases, the chance of being HD increases.
- Unfortunately we do not have a nice linear interpretation of β_1 over $Prob(HD = 1)$ anymore.

Interpretations through Plots

```
fram_data %>% mutate(HD = as.numeric(HD)-1) %>%
  ggplot(aes(x=SBP, y=HD)) +
  geom_jitter(height = .05, aes(color = factor(HD))) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
  # geom_smooth(method = "lm",    # may impose a liner model. we see the two curves are not the same.
  #             color = "red",
  #             se = FALSE) +
  ylab("Prob(HD=1)")
```

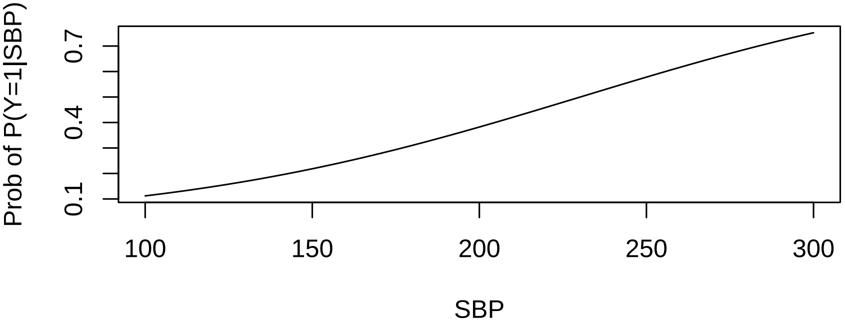
```
## `geom_smooth()` using formula 'y ~ x'
```



Interpretations through Plots

Alternatively, we can plot the prob through $\frac{e^{-3.66+0.0159\times SBP}}{1+e^{-3.66+0.0159\times SBP}}$

```
x <- seq(100, 300, by=1)
y <- exp(-3.66+0.0159*x)/(1+exp(-3.66+0.0159*x))
plot(x, y, pch=16, type = "l",
     xlab = "SBP",
     ylab = "Prob of P(Y=1|SBP)" )
```



Inference for the Coefficients

How can we tell if the true β_1 is not 0? We need to provide confidence intervals of hypotheses tests for the unknown parameters.

Usual z-intervals for the coefficient's (output from the summary)

```
confint.default(fit1)   # confint(fit1) Different test, namely Likelihood ratio tests
```

```
              2.5 %  97.5 %
(Intercept) -4.3416 -2.9852
SBP          0.0116  0.0202
```

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
    - Classification rules
    - Criterion for classifier
6. Multiple Logistic Regression and Classification
    - Multiple logistic regression
    - Final model
7. Classification Revisit
    - Two features
    - Final model
8. Training/Testing Data

Classification

- Given the prediction for the probability of Alice having heart disease, i.e. $\hat{P}(HD = 1|SBP = 100) = .11$, how do we decide whether Alice will have heart disease or not?
- In general, how do we classify $\hat{Y} = 1$ given $\hat{P}(Y = 1|X)$?
- A sensible way to predict $Y|x$ is to set a threshold over Prob(Y|x).

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Classification rules

Once we have an estimation equation for Prob(HD=1|SBP), we can immediately obtain prediction rules by setting threshold.

Rule 1: Thresholding probability by 1/2

By definition, the larger $\hat{P}(HD = 1|SBP)$ is, the more likely Alice will have heart disease. We start with a classifier that classifies $\widehat{HD} = 1$ if $\hat{P}(HD = 1|SBP) > 1/2$. To be more specific,

$$\widehat{HD} = 1 \quad \text{if} \quad \hat{P}(HD = 1|SBP) = \frac{e^{-3.66+.0159 \cdot SBP}}{1 + e^{-3.66+.0159 \cdot SBP}} > \frac{1}{2}.$$

```
ifelse(fit1.predict > 1/2, "1", "0")
```

```
## 1407
##  "0"
```

Thus we classify Alice as not having heart disease.

Linear Boundary

Linear boundary:

The classification rule above is equivalent to

$$\begin{aligned}\widehat{HD} = 1 \quad \text{if} \quad -3.66 + .0159 \cdot SBP &= \log(\text{odds ratio}) \\ &= \log\left(\frac{P(HD = 1|SBP)}{P(HD = 0|SBP)}\right) \\ &> \log\left(\frac{1/2}{1/2}\right) = \log 1 = 0\end{aligned}$$

Simple algebra yields

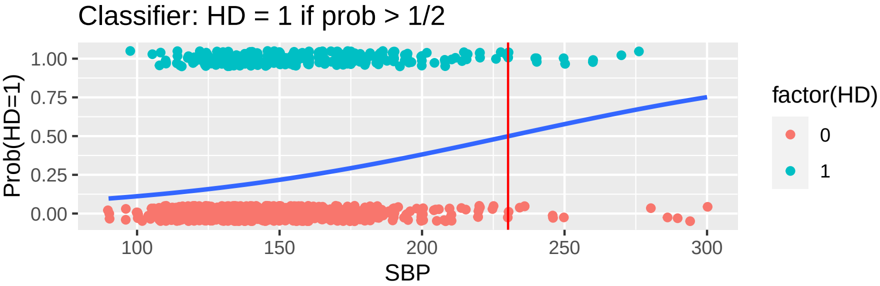
$$\hat{Y} = 1 \quad \text{if} \quad SBP > \frac{3.66}{.0159} = 230.18.$$

This is called the linear classification boundary with simple interpretation. Since Alice has $SBP = 100 < 230.18$, we classify Alice as not having heart disease.

Classifier: HD $= 1$ if prob $> 1/2$

```
fram_data %>% mutate(HD = as.numeric(HD)-1) %>%
  ggplot(aes(x=SBP, y=HD)) +
  geom_jitter(height = .05, aes(color = factor(HD))) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
  geom_vline(xintercept = 230.18, col="red") +
  ggtitle("Classifier: HD = 1 if prob > 1/2") +
  ylab("Prob(HD=1)")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Based on the above linear boundary, we classify everyone as $\hat{Y} = 1$ if their SBP is on the right side of the vertical line!

Classification rules

Rule 2: Thresholding probability by 1/3

Let's redo the exercise by thresholding $\hat{P}(HD = 1|SBP) > 1/3$.

$$\widehat{HD} = 1 \quad \text{if} \quad \hat{P}(HD = 1|SBP) = \frac{e^{-3.66+.0159 \cdot SBP}}{1 + e^{-3.66+.0159 \cdot SBP}} > \frac{1}{3}.$$

```
ifelse(fit1.predict > 1/3, "1", "0")
```

```
## 1407
##  "0"
```

We will still classify Alice as not having heart disease.

Classification rules

Alternatively, the linear classification rule is

$$\begin{aligned}\widehat{HD} = 1 \quad \text{if} \quad -3.66 + .0159 \cdot SBP &= \log(\text{odds ratio}) \\ &= \log\left(\frac{P(HD = 1|SBP)}{P(HD = 0|SBP)}\right) \\ &> \log\left(\frac{1/3}{2/3}\right) = \log(1/2) = -0.693\end{aligned}$$

Simple algebra yields

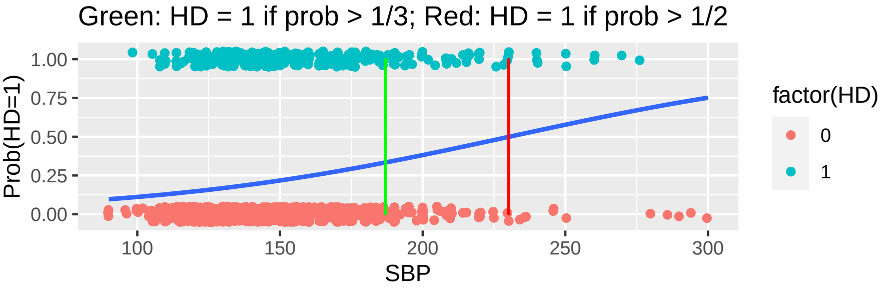
$$\hat{Y} = 1 \quad \text{if} \quad SBP > \frac{3.66 + \log(1/2)}{.0159} = 187$$

Comparison of Two Classifiers

We now compare the two classifiers

```
fram_data %>% mutate(HD = as.numeric(HD)-1) %>%
  ggplot(aes(x=SBP, y=HD)) +
  geom_jitter(height = .05, aes(color = factor(HD))) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
  geom_line(aes(x = 230.18), col="red") +
  geom_line(aes(x = 187), col="green") +
  ggtitle("Green: HD = 1 if prob > 1/3; Red: HD = 1 if prob > 1/2") +
  ylab("Prob(HD=1)")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Criterion for classifier

Given two classifiers, how can we tell which one is better?

Criterion: Misclassification errors

Misclassification error

Mean values of missclassifications

$$MCE = \frac{1}{n}\sum_{i=1}^{n}\{\hat{y}_i \neq y_i\}$$

```
fit1.pred.5 <- ifelse(fit1$fitted > 1/2, "1", "0")
error.training <- mean(fit1.pred.5 != fram_data$HD)
error.training
```

```
## [1] 0.223
```

```
accuracy <- 1 - error.training
accuracy
```

```
## [1] 0.777
```

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Multiple Logistic Regression and Classification

We have introduce elements of logistic regression models and classifications using only SBP. We can immediately extend all the concepts to include more possible risk factors.

For simplicity, we delete all cases with missing values. In general, this is not recommended!

```
summary(fram_data)
```

```
## HD            AGE            SEX            SBP            DBP        
## 0:1095   Min.   :45.0   FEMALE:737   Min.   : 90   Min.   : 50.0  
## 1: 311   1st Qu.:48.0   MALE  :669   1st Qu.:130   1st Qu.: 80.0  
##          Median :52.0                Median :142   Median : 90.0  
##          Mean   :52.4                Mean   :148   Mean   : 90.1  
##          3rd Qu.:56.0                3rd Qu.:160   3rd Qu.: 98.0  
##          Max.   :62.0                Max.   :300   Max.   :160.0  
##                                                                   
##      CHOL           FRW           CIG     
## Min.   : 96   Min.   : 52   Min.   : 0  
## 1st Qu.:200   1st Qu.: 94   1st Qu.: 0  
## Median :230   Median :103   Median : 0  
## Mean   :235   Mean   :105   Mean   : 8  
## 3rd Qu.:264   3rd Qu.:114   3rd Qu.:20  
## Max.   :430   Max.   :222   Max.   :60  
##               NA's   :11    NA's   :2   
```

```
fram_data.f <- na.omit(fram_data)
```

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Multiple logistic regression

Denote $x = (x_1, x_2, \ldots, x_p)$. The logit link for the full model is

$$logit(P(HD = 1|x)) = \log\left(\frac{P(HD = 1|x)}{P(HD = 0|x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

where

$$P(HD = 1|x) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots \beta_p x_p}}$$

Similarly,

- MLE's will be obtained through maximize the log likelihood function
- Wald tests/intervals hold for each coefficient

Logistic Regression Model with All Possible Features

We show next the logistic regression model with all possible features.

```
fit2 <- glm(HD~., fram_data.f, family=binomial)
summary(fit2)
```

```
##
## Call:
## glm(formula = HD ~ ., family = binomial, data = fram_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.705  -0.727  -0.556  -0.333   2.446
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.33480    1.03663   -9.00  < 2e-16 ***
## AGE          0.06249    0.01500    4.17  3.1e-05 ***
## SEXMALE      0.90610    0.15764    5.75  9.0e-09 ***
## SBP          0.01484    0.00389    3.82  0.00013 ***
## DBP          0.00288    0.00762    0.38  0.70594
## CHOL         0.00446    0.00151    2.96  0.00305 **
## FRW          0.00580    0.00406    1.43  0.15296
## CIG          0.01231    0.00609    2.02  0.04315 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.1  on 1385  degrees of freedom
## AIC: 1359
##
## Number of Fisher Scoring iterations: 4
```

Confidence Intervals

```
confint.default(fit2)
```

```
##                2.5 %   97.5 %
## (Intercept) -1.14e+01 -7.30304
## AGE          3.31e-02  0.09188
## SEXMALE      5.97e-01  1.21507
## SBP          7.22e-03  0.02246
## DBP         -1.21e-02  0.01781
## CHOL         1.51e-03  0.00741
## FRW         -2.15e-03  0.01374
## CIG          3.79e-04  0.02424
```

P(HD = 1|X)

The probability of $HD = 1$ given all the factors is estimated as

$$\hat{P}(HD = 1|X) = \frac{e^{-9.33480+0.06249\times AGE+0.90610 I_{Male}+\cdots+0.01231\times CIG}}{1 + e^{-9.33480+0.06249\times AGE+0.90610 I_{Male}+\cdots+0.01231\times CIG}}$$

Wald intervals

We conclude from the Wald test that DBP or FRW by itself is not significant.

```
Anova(fit2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: HD
##      LR Chisq Df Pr(>Chisq)
## AGE      17.6  1    2.7e-05 ***
## SEX      34.0  1    5.4e-09 ***
## SBP      14.7  1    0.00013 ***
## DBP       0.1  1    0.70599
## CHOL      8.7  1    0.00313 **
## FRW       2.0  1    0.15491
## CIG       4.0  1    0.04437 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
confint.default(fit2) #   predict(fit2,fram_data.new, type = "response" )
```

```
##                 2.5 %   97.5 %
## (Intercept) -1.14e+01 -7.30304
## AGE          3.31e-02  0.09188
## SEXMALE      5.97e-01  1.21507
## SBP          7.22e-03  0.02246
## DBP         -1.21e-02  0.01781
## CHOL         1.51e-03  0.00741
## FRW         -2.15e-03  0.01374
## CIG          3.79e-04  0.02424
```

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Final model

```
summary(glm(HD~AGE+SEX+SBP+CHOL+FRW+CIG, family=binomial, data=fram_data.f))
```

```
##
## Call:
## glm(formula = HD ~ AGE + SEX + SBP + CHOL + FRW + CIG, family = binomial,
##     data = fram_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.707  -0.728  -0.552  -0.334   2.450
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.22786    0.99615   -9.26  < 2e-16 ***
## AGE          0.06153    0.01478    4.16  3.1e-05 ***
## SEXMALE      0.91127    0.15712    5.80  6.6e-09 ***
## SBP          0.01597    0.00249    6.42  1.4e-10 ***
## CHOL         0.00449    0.00150    2.99   0.0028 **
## FRW          0.00604    0.00400    1.51   0.1315
## CIG          0.01228    0.00609    2.02   0.0437 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.3  on 1386  degrees of freedom
## AIC: 1357
##
## Number of Fisher Scoring iterations: 4
```

Final model - Backward Selection

As seen above, FRW is not significant so we can eliminate it.

```
fit.final.0 <- glm(HD~AGE+SEX+SBP+CHOL+FRW+CIG, family=binomial, data=fram_data.f)
summary(fit.final.0)
```

```
##
## Call:
## glm(formula = HD ~ AGE + SEX + SBP + CHOL + FRW + CIG, family = binomial,
##     data = fram_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.707  -0.728  -0.552  -0.334   2.450
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.22786    0.99615   -9.26  < 2e-16 ***
## AGE          0.06153    0.01478    4.16  3.1e-05 ***
## SEXMALE      0.91127    0.15712    5.80  6.6e-09 ***
## SBP          0.01597    0.00249    6.42  1.4e-10 ***
## CHOL         0.00449    0.00150    2.99   0.0028 **
## FRW          0.00604    0.00400    1.51   0.1315
## CIG          0.01228    0.00609    2.02   0.0437 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1343.3  on 1386  degrees of freedom
## AIC: 1357
##
## Number of Fisher Scoring iterations: 4
```

Final model

Note that FRW is not significant in fit.fial.0. Once we drop FRW (Backward Selection), all the remaining factors seem to be fine. (CIG shows weak evidence to be useful in this final model.) Nevertheless we decided to retain all the variables in the final model.

```
fit.final <- glm(HD~AGE+SEX+SBP+CHOL+CIG, family=binomial, data=fram_data.f)
summary(fit.final)
```

```
Call:
glm(formula = HD ~ AGE + SEX + SBP + CHOL + CIG, family = binomial,
    data = fram_data.f)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.754  -0.729  -0.554  -0.344   2.447

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.70228    0.92683   -9.39  < 2e-16 ***
AGE          0.06136    0.01475    4.16  3.2e-05 ***
SEXMALE      0.88575    0.15579    5.69  1.3e-08 ***
SBP          0.01709    0.00237    7.20  5.9e-13 ***
CHOL         0.00440    0.00150    2.93   0.0033 **
CIG          0.01136    0.00606    1.87   0.0608 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1469.3  on 1392  degrees of freedom
Residual deviance: 1345.5  on 1387  degrees of freedom
```

Findings from the Final model

So based on the study above, we may say collectively AGE, SBP, CHOL, CIG are all positively related to the chance of a HD while Male's have higher chance of HD comparing with Females controlling for all other factors in the model.

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Classification Revisit

The recipe for choosing a classifier with multiple logistic regression is similar to that using SBP alone.

- Fit the logistic regression
- Get a set of rules by thresholding the estimated probability as a function of the predictors
- Evaluate the performance of the set of rules using one criterion of your choice

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. Training/Testing Data

Two features

Before using the final model, let's use only SBP and AGE to take a look at the linear boundary.

Get the logit function and examine some specific rules

```
fit2 <- glm(HD~SBP+AGE, family= binomial, fram_data.f)
summary(fit2)
```

```
##
## Call:
## glm(formula = HD ~ SBP + AGE, family = binomial, data = fram_data.f)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.630  -0.721  -0.601  -0.466   2.169
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.48625    0.79182   -8.19  2.6e-16 ***
## SBP          0.01434    0.00225    6.38  1.8e-10 ***
## AGE          0.05775    0.01422    4.06  4.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1469.3  on 1392  degrees of freedom
## Residual deviance: 1400.8  on 1390  degrees of freedom
## AIC: 1407
##
## Number of Fisher Scoring iterations: 4
```

Two features

$$logit = -6.554 + 0.0144SBP + .0589AGE$$

Rule 1: threshold the probability at $2/3$

$$\begin{aligned}\widehat{HD} = 1 \text{ if } SBP &> \frac{-0.0589}{0.0144}Age + \frac{\log(2) + 6.554}{0.0144} \\ &= -4.09Age + 503\end{aligned}$$

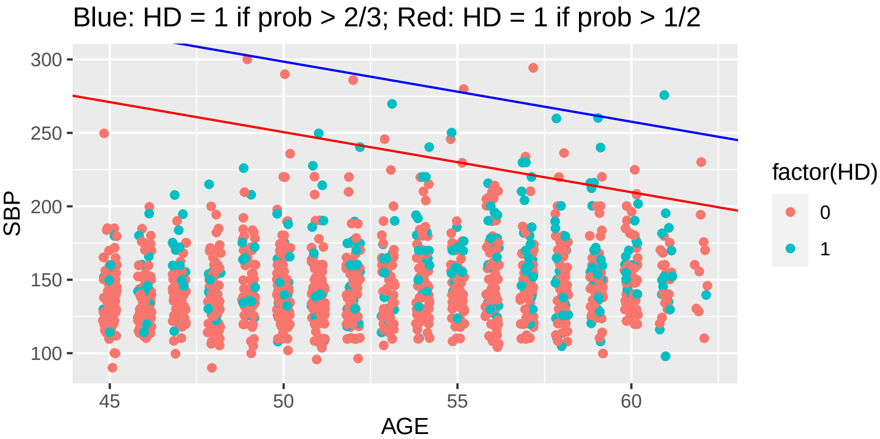
Rule 2: threshold the probability at $1/2$

$$\begin{aligned}\widehat{HD} = 1 \text{ if } SBP &> \frac{-0.0589}{0.0144}Age + \frac{\log(1) + 6.554}{0.0144} \\ &= -4.09Age + 455\end{aligned}$$

Two features

Let's put two linear boundaries together.

```
fram_data.f  %>%
  ggplot(aes(x=AGE, y=SBP)) +
  geom_jitter(width = 0.2, aes(color = factor(HD))) +
  geom_abline(intercept = 503, slope = -4.09, col = "blue") +
  geom_abline(intercept = 455, slope = -4.09, col = "red") +
  ggtitle("Blue: HD = 1 if prob > 2/3; Red: HD = 1 if prob > 1/2")
```



1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
    - Classification rules
    - Criterion for classifier
6. Multiple Logistic Regression and Classification
    - Multiple logistic regression
    - Final model
7. Classification Revisit
    - Two features
    - Final model
8. Training/Testing Data

Final model

Now we compare three models `fit1`, `fit2`, `fit.final` using misclassification errors as the criterion.

```
fit1.pred.5 <- ifelse(fit1$fitted > 1/2, "1", "0")
error.training.fit1 <- mean(fit1.pred.5 != fram_data$HD)
accuracy.fit1 <- 1 - error.training.fit1

fit2.pred.5 <- ifelse(fit2$fitted > 1/2, "1", "0")
error.training.fit2 <- mean(fit2.pred.5 != fram_data.f$HD)
accuracy.fit2 <- 1 - error.training.fit2

fit.final.pred.5 <- ifelse(fit.final$fitted > 1/2, "1", "0")
error.training.fit.final <- mean(fit.final.pred.5 != fram_data.f$HD)
accuracy.fit.final <- 1 - error.training.fit.final

error.training.fit1
```

```
## [1] 0.223
```

```
error.training.fit2
```

```
## [1] 0.22
```

```
error.training.fit.final
```

```
## [1] 0.217
```

Assessing Final model

- From misclassification errors, we can say that our final model is indeed better in terms of overall performance.
- After deciding on the model, we need to decide the classification rule depending on the goal.

Reflection

Remark

- Given a model, what threshold should we use so that the MCE will be minimized? The answer would be using 0.5 as the threshold number, assuming we treat each mistake the same.
- On the other hand, if the two types of the mistakes cost differently, we ought to weigh the type of the mistake.
- For example, if lower false negative is more important, then we can weigh more on false negative.
- A more concrete example is the disease diagnosis. Falsely diagnoses as disease-free is worse than falsely diagnosed as positive.

1. Objectives
2. Case Study: Framingham Heart Study
3. Logistic Regression: `HD~SBP`
4. Maximum Likelihood Estimators (MLE)
5. Classification
   - Classification rules
   - Criterion for classifier
6. Multiple Logistic Regression and Classification
   - Multiple logistic regression
   - Final model
7. Classification Revisit
   - Two features
   - Final model
8. **Training/Testing Data**

Training/Testing Data

Depending on the goal, we need to choose a criterion. Then we may use a testing data to find a good model. For a final model chosen we need a validation data to evaluate/report the performance.

We may split the data into three sub-samples.

- Training Data: fit a model
- Testing Data: compare models to find a best one
- Validation Data: to evaluate the final model

Choose the model by some criterion. For example, the one with the lowest misclassification error.

Split Data into Training, Testing, and Validation Data

We first split the data into `data.train`, `data.test` and `data.val`.

```r
# Split the data:
N <- length(fram_data.f$HD)
n1 <- floor(.6*N)
n2 <- floor(.2*N)

set.seed(10)
# Split data to three portions of .6, .2 and .2 of data size N

idx_train <- sample(N, n1)
idx_no_train <- (which(! seq(1:N) %in% idx_train))
idx_test <- sample( idx_no_train, n2)
idx_val <- which(! idx_no_train %in% idx_test)
data.train <- fram_data.f[idx_train,]
data.test <- fram_data.f[idx_test,]
data.val <- fram_data.f[idx_val,]
```

Training Models

Fit two models using `data.train`

```
fit1.train <- glm(HD~SBP, data=data.train, family=binomial)
summary(fit1.train)
```

```
##
## Call:
## glm(formula = HD ~ SBP, family = binomial, data = data.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.577  -0.726  -0.648  -0.556   2.032
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.30459    0.45226   -7.31  2.7e-13 ***
## SBP          0.01403    0.00292    4.81  1.5e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 905.28  on 834  degrees of freedom
## Residual deviance: 881.74  on 833  degrees of freedom
## AIC: 885.7
##
## Number of Fisher Scoring iterations: 4
```

```
fit5.train <- glm(HD~SBP+SEX+AGE+CHOL+CIG, data=data.train, family=binomial)
summary(fit5.train)
```

```
##
```

Training Models

Fit two models using data.train

```
fit5.train <- glm(HD~SBP+SEX+AGE+CHOL+CIG, data=data.train, family=binomial)
summary(fit5.train)
```

```
##
## Call:
## glm(formula = HD ~ SBP + SEX + AGE + CHOL + CIG, family = binomial,
##     data = data.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.651  -0.747  -0.572  -0.335   2.345
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.83429    1.17063   -7.55  4.5e-14 ***
## SBP          0.01480    0.00314    4.72  2.4e-06 ***
## SEXMALE      0.92275    0.19388    4.76  1.9e-06 ***
## AGE          0.07210    0.01862    3.87  0.00011 ***
## CHOL         0.00428    0.00187    2.29  0.02200 *
## CIG          0.01112    0.00775    1.43  0.15158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 905.28  on 834  degrees of freedom
## Residual deviance: 829.94  on 829  degrees of freedom
## AIC: 841.9
##
## Number of Fisher Scoring iterations: 4
```

Fitted Probabilities Using Testing Data

Get the fitted probabilities using the testing data

```
fit1.fitted.test <- predict(fit1.train, data.test, type="response") # get the prob's
fit5.fitted.test <- predict(fit5.train, data.test, type="response")

data.frame(fit1.fitted.test, fit5.fitted.test)[1:10, ]
```

```
##      fit1.fitted.test fit5.fitted.test
## 252             0.177            0.176
## 88              0.203            0.226
## 1110            0.285            0.276
## 570             0.291            0.406
## 952             0.185            0.090
## 621             0.198            0.357
## 1269            0.285            0.180
## 550             0.181            0.424
## 931             0.169            0.108
## 651             0.405            0.626
```

Look at the first 10 rows. Notice that row names are the subject numbers chosen in the testing data. The estimated probability for each row is different using `fit1` and `fit5`.

MCE for the Two Models

Suprisingly, the two models have the same misclassification errors using the testing data.

```
fit1.test.pred.5 <- ifelse(fit1.fitted.test > 1/2, "1", "0")
error.testing.fit1.final <- mean(fit1.test.pred.5 != data.test$HD)

fit5.test.pred.5 <- ifelse(fit5.fitted.test > 1/2, "1", "0")
error.testing.fit5.final <- mean(fit5.test.pred.5 != data.test$HD)

sum(fit1.test.pred.5!=fit5.test.pred.5)
```

```
## [1] 12
```

```
error.testing.fit1.final
```

```
## [1] 0.241
```

```
error.testing.fit5.final
```

```
## [1] 0.241
```

Remark

Remark: On a separate issue, there are variabilities on results as splitting process. If you comment out set.seed and repeat running the above training/testing/validation session our final output will be different.

Pipeline functions

```
# Get the design matrix without 1's and HD
Xy_design <- model.matrix(HD ~.+0, fram_data.f)
# Attach y as the last column.
Xy <- data.frame(Xy_design, fram_data.f$HD)

fit.all <- bestglm(Xy, family = binomial, method = "exhaustive", IC="AIC", nvmax = 10) # method = "exhaustive",
```

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

```
# Split the data:
N <- length(fram_data.f$HD)
n1 <- floor(.6*N)
n2 <- floor(.2*N)

set.seed(10)
# Split data to three portions of .6, .2 and .2 of data size N

idx_train <- sample(N, n1)
idx_no_train <- (which(! seq(1:N) %in% idx_train))
idx_test <- sample( idx_no_train, n2)
idx_val <- which(! idx_no_train %in% idx_test)
data.train <- fram_data.f[idx_train,]
data.test <- fram_data.f[idx_test,]
data.val <- fram_data.f[idx_val,]
```

Appendix

Confusion matrix

Given the actual status and the predicted status by a classifier, we can summarize how well this rule works by a two-way table which we called `confusion matrix`.

Let's use our first classifier,

$$\widehat{HD} = 1 \text{ if } \hat{P}(HD = 1|SBP) > 1/2.$$

```
fit1.pred.5 <- ifelse(fit1$fitted > 1/2, "1", "0")
```

Confusion matrix

Take a few participants in the data, we compare the facts and the predictions:

```
set.seed(10)
output1 <- data.frame(fram_data$HD, fit1.pred.5, fit1$fitted)[sample(1406, 10),]
names(output1) <- c( "HD", "Predicted HD", "Prob")
output1
```

```
##      HD Predicted HD  Prob
## 491   0            0 0.192
## 1354  0            0 0.168
## 368   1            0 0.168
## 439   0            0 0.147
## 344   0            0 0.261
## 1295  0            0 0.435
## 143   0            0 0.246
## 938   0            0 0.158
## 1405  1            0 0.223
## 930   0            0 0.151
```

We see there are 4 participants mislabeled.

Confusion matrix

A 2 by 2 table which summarize the number of mis/agreed labels

```
cm.5 <- table(fit1.pred.5, fram_data$HD)
cm.5
```

```
##
## fit1.pred.5    0    1
##           0 1084  302
##           1   11    9
```

Note that the rows are $\hat{y}$ and the columns are y. These four numbers reflects different criteria.

	$Y = 0$	$Y = 1$
$\hat{Y} = 0$	Specificity	False negative
$\hat{Y} = 1$	False positive	Sensitivity (true positive)

Sensitivity and Specificity

Sensitivity:

The sensitivity is defined as

$$P(\hat{Y} = 1 | Y = 1)$$

This is also called `True Positive Rate`: the proportion of correct positive classification.

```
sensitivity <- cm.5[2,2]/sum(cm.5[,2])
sensitivity
```

```
## [1] 0.0289
```

Sensitivity and Specificity

Specificity:

The specificity is defined as

$$P(\hat{Y} = 0 | Y = 0)$$

`Specificity` measures the proportion of correct negative classification.

```
specificity <- cm.5[1,1]/sum(cm.5[,1])
specificity
```

```
## [1] 0.99
```

Sensitivity and Specificity

False Positive:

A related measure is `false positive rate`.

$$1 - \text{Specificity} = P(\hat{Y} = 1 | Y = 0)$$

`False Positive` measures the proportion of incorrect positive classification (given the actual status being negative).

```
false.positive <- cm.5[2,1]/sum(cm.5[,1])
false.positive
```

```
## [1] 0.01
```

ROC curve and AUC

ROC curve

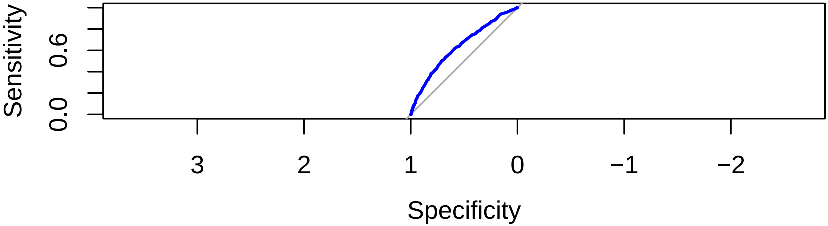
- Given a logistic model, we can obtain different classifiers by changing the classification rule, i.e. by changing the threshold. Each classifier has its sensitivity and specificity. We can set sensitivity as x-axis and specificity as y-axis and plot all the pairs of sensitivity and specificity. This curve is termed as `ROC curve`.
- `ROC curve` is helpful when choosing classifiers. We want to have both high specificity and high sensitivity at the same time, which means we want to classify both $Y = 0$ and $Y = 1$ correctly. However, in general, we will NOT have a perfect classifier and need to strive a balance between the two.
- We use the `roc()` function from package `pROC` to obtain detailed information for each classifier. Notice the ROC curve here is Sensitivity vs. Specificity. Some prefer using false positive as x-axis so as to have an ascending x-axis.

ROC curve

```
fit1.roc<- roc(fram_data$HD, fit1$fitted, plot=T, col="blue")
```

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

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

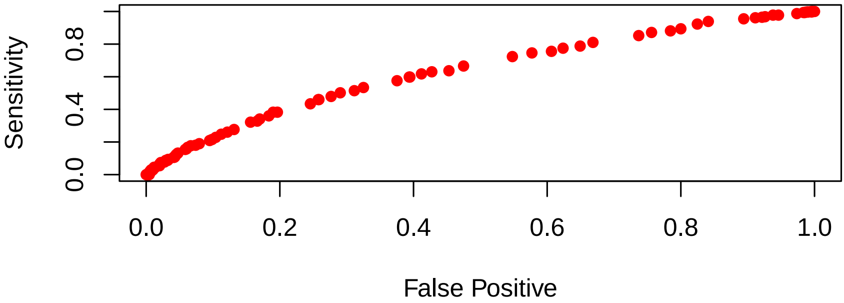


Note that the higher the specificity, the lower the sensitivity. A perfect classifier will have both Specificity = 1 and Sensitivity = 1.

ROC curve

Compare the following ROC curve using false positive as x-axis.

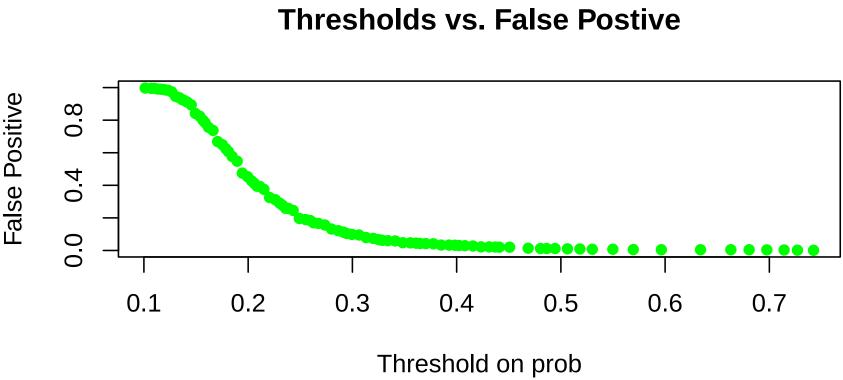
```
plot(1-fit1.roc$specificities, fit1.roc$sensitivities, col="red", pch=16,
     xlab="False Positive",
     ylab="Sensitivity")
```



ROC curve

We can also plot a curve that shows the probability thresholds used and the corresponding False Positive rate.

```
plot(fit1.roc$thresholds, 1-fit1.roc$specificities,  col="green", pch=16,
     xlab="Threshold on prob",
     ylab="False Positive",
     main = "Thresholds vs. False Postive")
```



AUC (Area under the curve)

AUC measures the area under the ROC curve. It is used to measure the performance of the logistic model as a whole: the larger the better. Why?

Given a specificity, the model with a higher sensitivity is more desirable. If a model has higher sensitivity for each specificity, this is a better model. An extreme case is when a model has sensitivity being constantly 1 and thus $AUC = 1$.

```
fit1.roc$auc
pROC::auc(fit1.roc)
### if you get "Error in rank(prob) : argument "prob" is missing, with no default"
### then it is possible that you are using the auc() in glmnet
### use pROC::auc(fit1.roc) to specify we want to use auc() in pROC
```

```
## Area under the curve: 0.637
## Area under the curve: 0.637
```

Misclassification error

Mean values of missclassifications

$$MCE = \frac{1}{n}\sum_{i=1}^{n}\{\hat{y}_i \neq y_i\}$$

```
error.training <- mean(fit1.pred.5 != fram_data$HD)
error.training
```

```
## [1] 0.223
```

```
accuracy <- 1 - error.training
accuracy
```

```
## [1] 0.777
```

False Discovery Rate (FDR)

False discovery rate measures the expected proportion of false discovery (incorrectly reject the null hypotheses).

$\mathrm{FDR} = P(Y = 0 | \hat{Y} = 1)$

```
fdr <- cm.5[2,1] / sum(cm.5[2,])
```

False Discovery Rate (FDR)

Two related concepts are true positive and true negative.

Positive Prediction (true positive):

Positive Prediction is a measure of the accuracy given the predictions.

Positive Prediction $= P(Y = 1|\hat{Y} = 1)$

```r
positive.pred <- cm.5[2,2] / sum(cm.5[2,])
positive.pred
```

```
## [1] 0.45
```

Negative Prediction:

Negative Prediction $= P(Y = 0|\hat{Y} = 0)$

```r
negative.pred <- cm.5[1,1] / sum(cm.5[1,])
negative.pred
```

```
## [1] 0.782
```