# Statistical Modeling
# BANA 7042

Lecture 3: Logistic regression model - II

Dr. Dungang Liu

# Western Collaborative Group Study

- One of the earliest studies that addressed the heart disease issue.
- Started in 1960.
- 3154 healthy men:
  - Aged from 39 to 59;
  - From the San Francisco area;
  - Free of heart disease at the start of the study.


- 8.5 years later, the study recorded whether these men suffered from heart disease along with many other variables that might be related.
- Rosenman et al. (1975), JAMA, 233(8), 872-877.

# So far, we have learnt ....

- How to visualize binary data? (pie charts and histograms)
- How to visualize the association between a binary response and a covariate? (X-Y plots, interleaved histograms)
- Why a linear regression model may not be appropriate for modeling a binary response?
- How do we extend the idea of linear regression to the situation of a binary response?
- How do we do prediction given a fitted logistic regression model?
- How do we interpret the coefficients? Or how can we explain the impact of a predictor on the binary response?
- How do we estimate the coefficients?

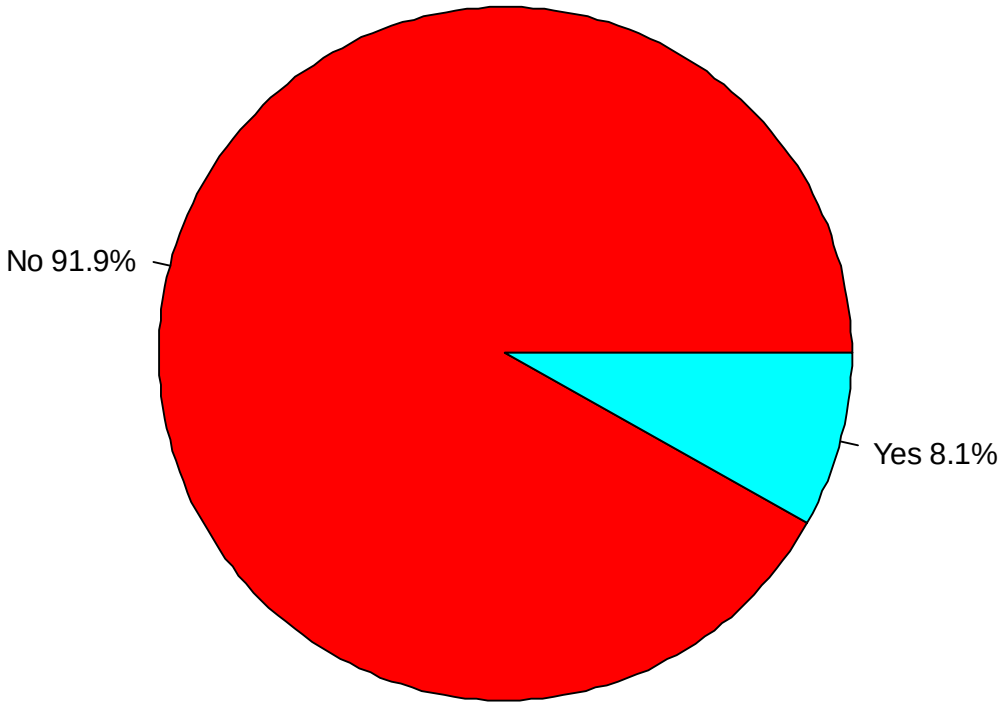
# Load the data set “wcgs”

```
library("faraway")

data(wcgs)
str(wcgs)
```

# Focusing on 3 variables

```
summary(wcgs[,c("chd","height","cigs")])
 chd          height          cigs      
no :2897   Min.   :60.00   Min.   : 0.0  
yes: 257   1st Qu.:68.00   1st Qu.: 0.0  
           Median :70.00   Median : 0.0  
           Mean   :69.78   Mean   :11.6  
           3rd Qu.:72.00   3rd Qu.:20.0  
           Max.   :78.00   Max.   :99.0  
```

**Pie chart of Coronary Heart Disease**



# Fit a logistic regression model

```
lmod<-glm(chd~height+cigs,family=binomial,wcgs)
summary(lmod)

Call:
glm(formula = chd ~ height + cigs, family = binomial, data = wcgs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0041  -0.4425  -0.3630  -0.3499   2.4357

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.50161    1.84186  -2.444   0.0145 *
height       0.02521    0.02633   0.957   0.3383
cigs         0.02313    0.00404   5.724 1.04e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1781.2  on 3153  degrees of freedom
Residual deviance: 1749.0  on 3151  degrees of freedom
AIC: 1755

Number of Fisher Scoring iterations: 5
```

# Model inference

- Which variables are "significant"? Do we need to include "height" in the model?
  - Model-L: chd ~ height + cigs  Likelihood function $L_L$
  - Model-S: chd ~ cigs  Likelihood function $L_S$

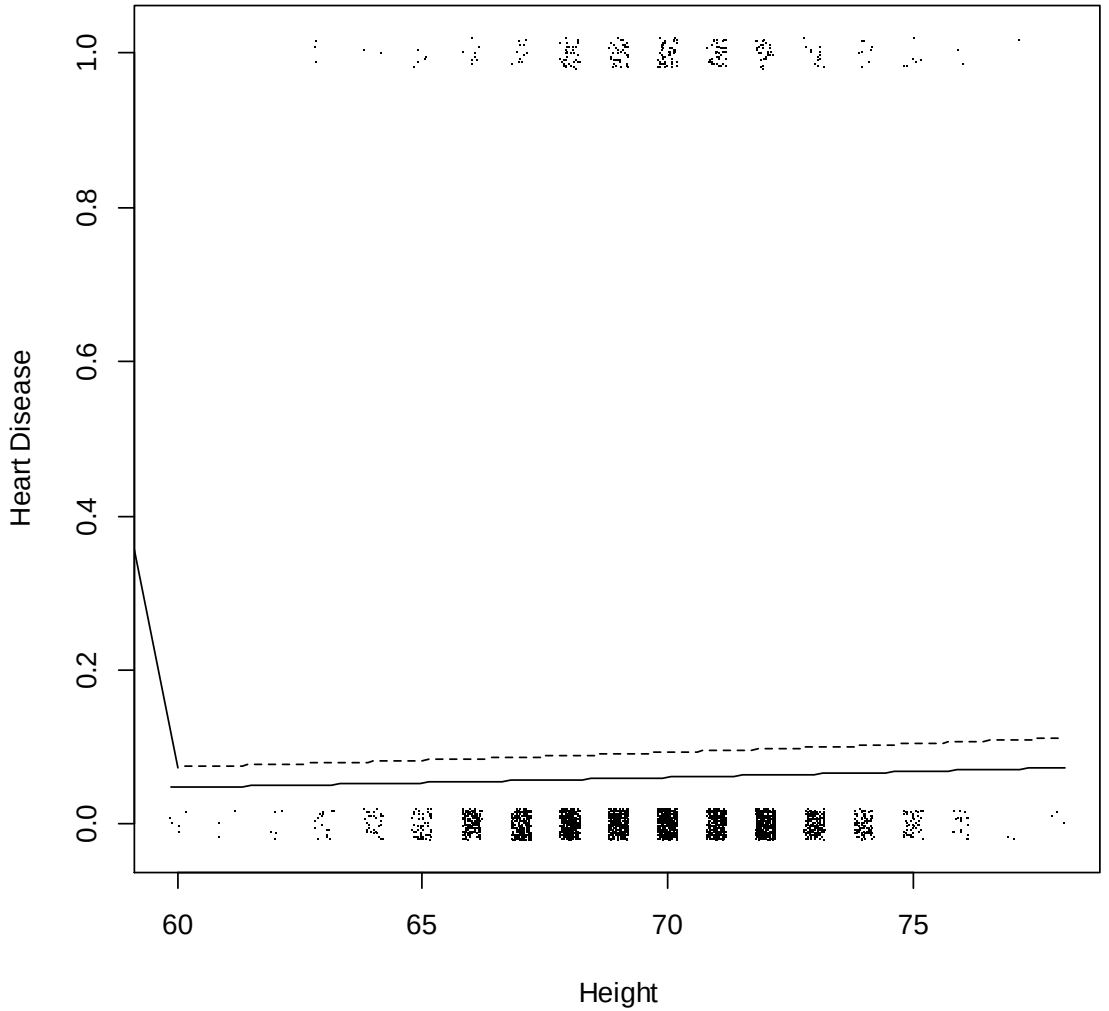
- If the smaller model is right,

$$2 \log \frac{L_L}{L_S} = D_L - D_S \sim \chi^2$$

# Please try …

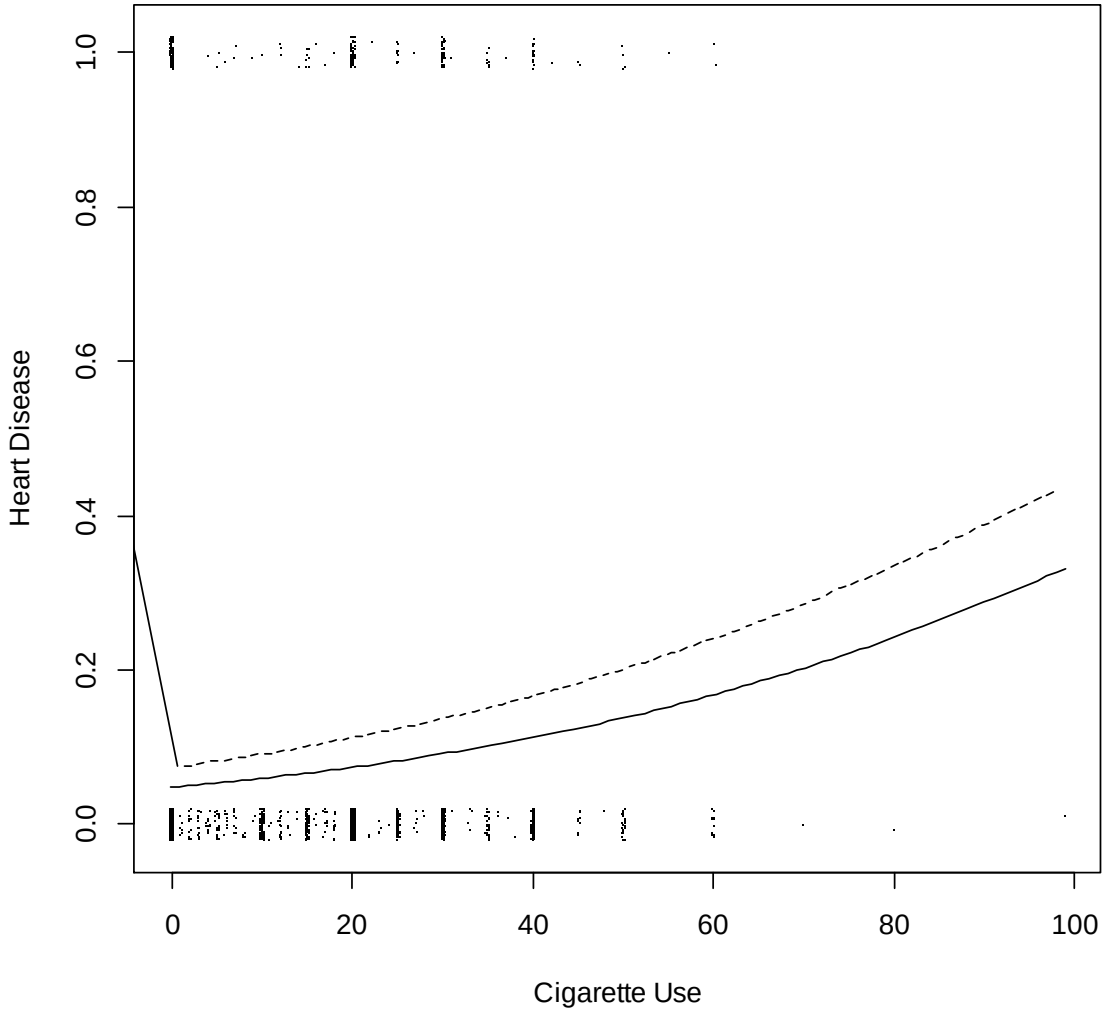
```
### Compare the models with and without “height”
lmod.s<-glm(chd~cigs,family=binomial,wcgs)
anova(lmod.s,lmod,test="Chi")

### Drop one covariate each time
drop1(lmod,test="Chi")
```

```
> anova(lmod.s,lmod,test="Chi")
Analysis of Deviance Table

Model 1: chd ~ cigs
Model 2: chd ~ height + cigs
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      3152       1750
2      3151       1749  1  0.92025   0.3374

> drop1(lmod,test="Chi")
Single term deletions

Model:
chd ~ height + cigs
       Df Deviance    AIC     LRT Pr(>Chi)
<none>      1749.0 1755.0
height  1   1750.0 1754.0  0.9202   0.3374
cigs    1   1780.1 1784.1 31.0695 2.49e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```


Heart Disease
0.0
0.2
0.4
0.6
0.8
1.0
60
65
70
75
Height

```
plot(jitter(y,0.1)~jitter(cigs),wcgs,xlab="Cigarette Use",ylab="Heart Disease",pch=".")
curve(ilogit(beta.lmod[1]+beta.lmod[2]*60+beta.lmod[3]*x),add=TRUE)
curve(ilogit(beta.lmod[1]+beta.lmod[2]*78+beta.lmod[3]*x),lty=2,add=TRUE)
```

**Question**: how to interpret this plot?



# Variable (Model) Selection

age

age in years

height

height in inches

weight

weight in pounds

sdp

systolic blood pressure in mm Hg

dbp

diastolic blood pressure in mm Hg

chol

Fasting serum cholesterol in mm %

behave

behavior type which is a factor with levels A1 A2 B3 B4

cigs

number of cigarettes smoked per day

dibep

behavior type a factor with levels A (Agressive) B (Passive)

chd

coronary heat disease developed is a factor with levels no yes

typechd

type of coronary heart disease is a factor with levels angina infdeath none silent

timechd

Time of CHD event or end of follow-up

arcus

arcus senilis is a factor with levels absent present

# Begins with a full model

```
### Add a derived new variable "BMI"
wcgs$bmi<-with(wcgs,703*wcgs$weight/(wcgs$height^2))

### A model with all variables available
lmod.full<-
glm(chd~age+height+weight+bmi+sdp+dbp+chol+dibep+cigs+
arcus,family=binomial,wcgs)

summary(lmod.full)
```

```
glm(formula = chd ~ age + height + weight + bmi + sdp + dbp +
    chol + dibep + cigs + arcus, family = binomial, data = wcgs)


Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -20.940481  16.069445  -1.303  0.19253
age            0.061621   0.012397   4.970 6.68e-07 ***
height         0.121465   0.228779   0.531  0.59547
weight        -0.014153   0.045136  -0.314  0.75385
bmi            0.159691   0.315675   0.506  0.61295
sdp            0.018227   0.006415   2.841  0.00449 **
dbp           -0.001139   0.010884  -0.105  0.91662
chol           0.010736   0.001532   7.007 2.44e-12 ***
dibepB         0.658080   0.145958   4.509 6.52e-06 ***
cigs           0.020985   0.004292   4.890 1.01e-06 ***
arcuspresent   0.209634   0.143874   1.457  0.14510
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1769.2  on 3139  degrees of freedom
Residual deviance: 1569.2  on 3129  degrees of freedom
  (14 observations deleted due to missingness)
AIC: 1591.2

Number of Fisher Scoring iterations: 6
```

# Check the signs (directions)!

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -20.940481  16.069445  -1.303  0.19253
age           0.061621   0.012397   4.970 6.68e-07 ***
height        0.121465   0.228779   0.531  0.59547
weight       -0.014153   0.045136  -0.314  0.75385
bmi           0.159691   0.315675   0.506  0.61295
sdp           0.018227   0.006415   2.841  0.00449 **
dbp          -0.001139   0.010884  -0.105  0.91662
chol          0.010736   0.001532   7.007 2.44e-12 ***
dibepB        0.658080   0.145958   4.509 6.52e-06 ***
cigs          0.020985   0.004292   4.890 1.01e-06 ***
arcuspresent  0.209634   0.143874   1.457  0.14510
```

# Check collinearity!

```
attach(wcgs)
round(cor(data.frame(height,weight,bmi)),1)
round(cor(sdp,dbp),1)
```

# What variables should we select to represent a group?

```
> round(cor(data.frame(height,weight,bmi)),1)
       height weight  bmi
height    1.0    0.5 -0.1
weight    0.5    1.0  0.8
bmi      -0.1    0.8  1.0

> round(cor(sdp,dbp),1)
[1] 0.8
```

# Compare two marginal models

```
summary(glm(chd~sdp,family=binomial))
summary(glm(chd~dbp,family=binomial))
```

```
> summary(glm(chd~sdp,family=binomial))


Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.926461   0.497037 -11.924  < 2e-16 ***
sdp          0.026671   0.003671   7.265 3.73e-13 ***
---

AIC: 1736.4


> summary(glm(chd~dbp,family=binomial))


Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.221713   0.511646 -10.206  < 2e-16 ***
dbp          0.033560   0.005981   5.611 2.01e-08 ***
---
AIC: 1755.7
```

```
### Use(height+bmi) to represent (height+weight+bmi)
### Use(dbp) to represent (sdp + dbp )
### Fit a new full model
lmod.full2<-
glm(chd~age+height+bmi+dbp+chol+dibep+cigs+arcus,family=binomial,wcgs)

summary(lmod.full2)
```

```
glm(formula = chd ~ age + height + bmi + dbp + chol + dibep +
    cigs + arcus, family = binomial, data = wcgs)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -15.677176   2.269985  -6.906 4.97e-12 ***
age            0.064941   0.012302   5.279 1.30e-07 ***
height         0.049734   0.027660   1.798  0.07217 .
bmi            0.062367   0.027357   2.280  0.02262 *
dbp            0.022086   0.007095   3.113  0.00185 **
chol           0.010760   0.001518   7.088 1.36e-12 ***
dibepB         0.662243   0.145689   4.546 5.48e-06 ***
cigs           0.022206   0.004255   5.219 1.80e-07 ***
arcuspresent   0.209776   0.143572   1.461  0.14398
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1769.2  on 3139  degrees of freedom
Residual deviance: 1577.2  on 3131  degrees of freedom
  (14 observations deleted due to missingness)
AIC: 1595.2
```

# Model selection based on p-values

1. Start with the full model with all the available predictors.
2. Compare this model with all the models with one less predictor. Computing the p-value corresponding to each dropped predictor. The "drop1" function in R can be used for this purpose.
3. Remove the variable with the largest p-value that is greater than some preset threshold, say 0.05.
4. Repeat the cycle 1-3 until no variable will be removed.

Please try: drop1(lmod.full2,test="Chi")

```
> drop1(lmod.full2,test="Chi")
Single term deletions

Model:
chd ~ age + height + bmi + dbp + chol + dibep + cigs + arcus
       Df Deviance    AIC    LRT  Pr(>Chi)
<none>      1577.2 1595.2
age     1   1605.0 1621.0 27.814 1.336e-07 ***
height  1   1580.5 1596.5  3.259  0.071013 .
bmi     1   1582.3 1598.3  5.130  0.023516 *
dbp     1   1586.6 1602.6  9.416  0.002151 **
chol    1   1628.7 1644.7 51.533 7.041e-13 ***
dibep   1   1598.9 1614.9 21.696 3.194e-06 ***
cigs    1   1603.4 1619.4 26.222 3.044e-07 ***
arcus   1   1579.3 1595.3  2.110  0.146365
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Model zero

```
> lmod.0<-glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial,wcgs)
> summary(lmod.0)

Call:
glm(formula = chd ~ age + height + bmi + dbp + chol + dibep +
    cigs, family = binomial, data = wcgs)


Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.070539   2.259732  -7.112 1.15e-12 ***
age           0.067982   0.012016   5.658 1.53e-08 ***
height        0.054814   0.027516   1.992  0.04636 *
bmi           0.057231   0.027295   2.097  0.03601 *
dbp           0.022378   0.007034   3.181  0.00147 **
chol          0.011107   0.001510   7.357 1.87e-13 ***
dibepB        0.658159   0.144974   4.540 5.63e-06 ***
cigs          0.022112   0.004239   5.216 1.83e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1779.2  on 3141  degrees of freedom
Residual deviance: 1588.0  on 3134  degrees of freedom
  (12 observations deleted due to missingness)
AIC: 1604
```

# Model selection based on AIC

- The Akaike Information criterion (AIC) is a popular way of choosing a model. The AIC value is defined

$$\text{AIC} = -2\log L + 2q$$

  where $L$ is the likelihood function and $q$ is the number of variables.
- We select the model with the **smallest** AIC value.

Please try:

```
model.AIC<-step(lmod.0,trace=0)
summary(model.AIC)
```

# Model.AIC is the same as Model zero

```
> summary(model.AIC)


Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.070539   2.259732  -7.112 1.15e-12 ***
age           0.067982   0.012016   5.658 1.53e-08 ***
height        0.054814   0.027516   1.992  0.04636 *
bmi           0.057231   0.027295   2.097  0.03601 *
dbp           0.022378   0.007034   3.181  0.00147 **
chol          0.011107   0.001510   7.357 1.87e-13 ***
dibepB        0.658159   0.144974   4.540 5.63e-06 ***
cigs          0.022112   0.004239   5.216 1.83e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1779.2  on 3141  degrees of freedom
Residual deviance: 1588.0  on 3134  degrees of freedom
  (12 observations deleted due to missingness)
AIC: 1604
```

# As compared to …

```
> model.AIC2<-step(lmod.full,trace=0)
> summary(model.AIC2)


Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -15.957599   2.286076  -6.980 2.94e-12 ***
age           0.061590   0.012397   4.968 6.76e-07 ***
height        0.050161   0.027824   1.803   0.0714 .
bmi           0.060385   0.026599   2.270   0.0232 *
sdp           0.017728   0.004155   4.267 1.98e-05 ***
chol          0.010709   0.001529   7.006 2.45e-12 ***
dibepB        0.657616   0.145898   4.507 6.56e-06 ***
cigs          0.021041   0.004262   4.936 7.96e-07 ***
arcuspresent  0.210998   0.143718   1.468   0.1421
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1769.2  on 3139  degrees of freedom
Residual deviance: 1569.3  on 3131  degrees of freedom
  (14 observations deleted due to missingness)
AIC: 1587.3
```

# Model evaluation based on predictive power

```
wcgsm<-na.omit(wcgs) ### Remove the rows with missing values
dim(wcgsm)
model.final<-
glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial,wcgsm)
linpred<-predict(model.final)  ### Linear predictor
predprob<-predict(model.final,type="response")  ### predicted
probabilities
predout<-ifelse(predprob<0.5,"no","yes")  ### Predicted outcomes using
0.5 as the threshold
wcgsm<-data.frame(wcgsm,predprob,predout)
xtabs(~chd+predout,wcgsm)
```

# Specificity and Sensitivity

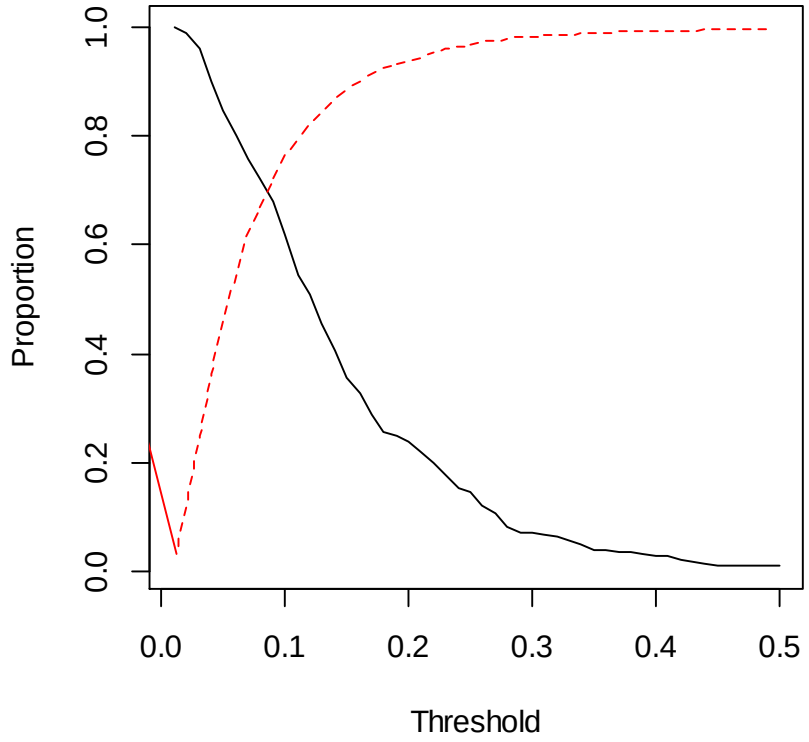
```
> xtabs(~chd+predout,wcgsm)
     predout
chd      no  yes
  no   2880    5
  yes   253    2
```

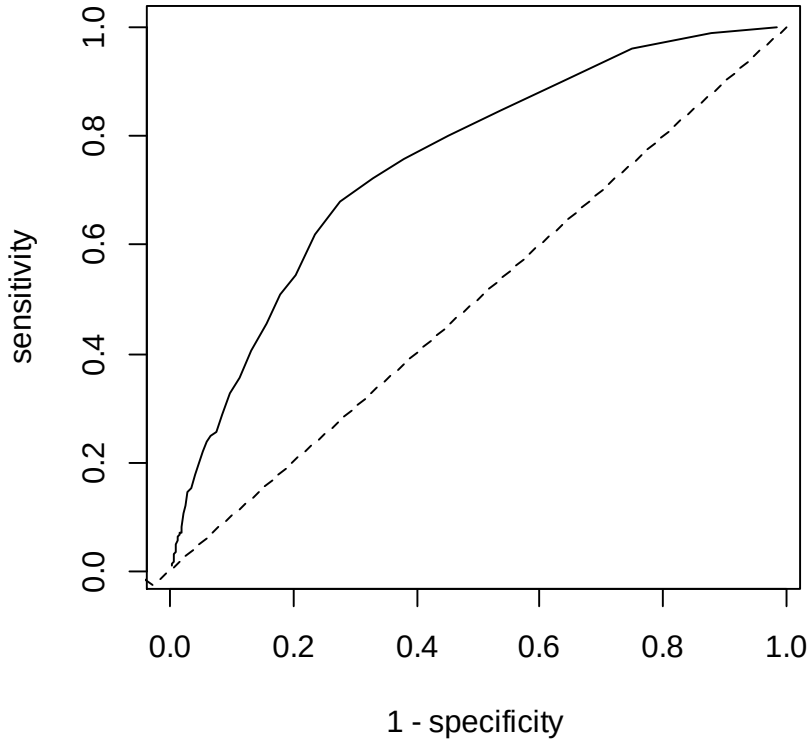
Specificity: 2880/(2880+5)=99.8%

Sensitivity: 2/(2+253)=0.8%

# Varying the threshold

```
thresh<-seq(0.01,0.5,0.01)
sensitivity<-specificity<-rep(NA,length(thresh))
for(j in seq(along=thresh)){
  pp<-ifelse(wcgsm$predprob<thresh[j],"no","yes")
  xx<-xtabs(~chd+pp,wcgsm)
  specificity[j]<-xx[1,1]/(xx[1,1]+xx[1,2])
  sensitivity[j]<-xx[2,2]/(xx[2,1]+xx[2,2])
}

par(mfrow=c(1,2))
matplot(thresh,cbind(sensitivity,specificity),type="l",xlab="Thr
eshold",ylab="Proportion",lty=1:2)
plot(1-specificity,sensitivity,type="l");abline(0,1,lty=2)
```


Proportion
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.1
0.2
0.3
0.4
0.5
Threshold


sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1 - specificity

# Make prediction with confidence

- Given some specific values of the predictors, how can we predict the chance of developing heart disease with certain **confidence**?

- For example, given an individual (age=50, height=70, bmi=25, dbp=80, chol=200, dibep="B",cigs=20), can we have an interval such that with 95% confidence that his “true” chance of HD will fall within?

predict.glm {stats} R Documentation

## Predict Method for GLM Fits

### Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.

### Usage

```
## S3 method for class 'glm'
predict(object, newdata = NULL,
            type = c("link", "response", "terms"),
            se.fit = FALSE, dispersion = NULL, terms = NULL,
            na.action = na.pass, ...)
```

### Arguments

`object`
: a fitted object of class inheriting from `"glm"`.

`newdata`
: optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.

`type`
: the type of prediction required. The default is on the scale of the linear predictors; the alternative `"response"` is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and `type = "response"` gives the predicted probabilities. The `"terms"` option returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale.

  The value of this argument can be abbreviated.

`se.fit`
: logical switch indicating if standard errors are required.

# Please try …

```
new.ind<-
data.frame(age=50,height=70,bmi=25,dbp=80,chol=200,dibep="B",cigs=20)

### Predict the value
predict(lmod.0,newdata=new.ind,type="link")  ### Linear predictor
predict(lmod.0,newdata=new.ind,type="response") ### Probability


### Predict the value with its standard error
predict(lmod.0,newdata=new.ind,type="link",se=T)
predict(lmod.0,newdata=new.ind,type="response",se=T)
```

```
> predict(lmod.0,newdata=new.ind,type="link",se=T)
$fit
        1
-2.291606

$se.fit
[1] 0.1156408

> round(ilogit(c(-2.2916-1.96*0.1156,-2.2916+1.96*0.1156)),3)  ### Confidence interval
[1] 0.075 0.113



> predict(lmod.0,newdata=new.ind,type="response",se=T)
$fit
         1
0.09182058

$se.fit
          1
0.009643238

> round(c(0.09182-1.96*0.00964,0.09182+1.96*0.00964),3) ### Confidence interval
[1] 0.073 0.111
```

# Variations of logistic regression

# How can we extend the idea to the case of a binary response?

- Suppose the binary response

$$Y \sim \text{Bernoulli}(p)$$

- We will make two additional assumptions on top of this assumption:
  1. Y relies on X only through its mean, e.g $Y \sim \text{Bernoulli}(p(X))$
  2. The logit transformation of the parameter $p$

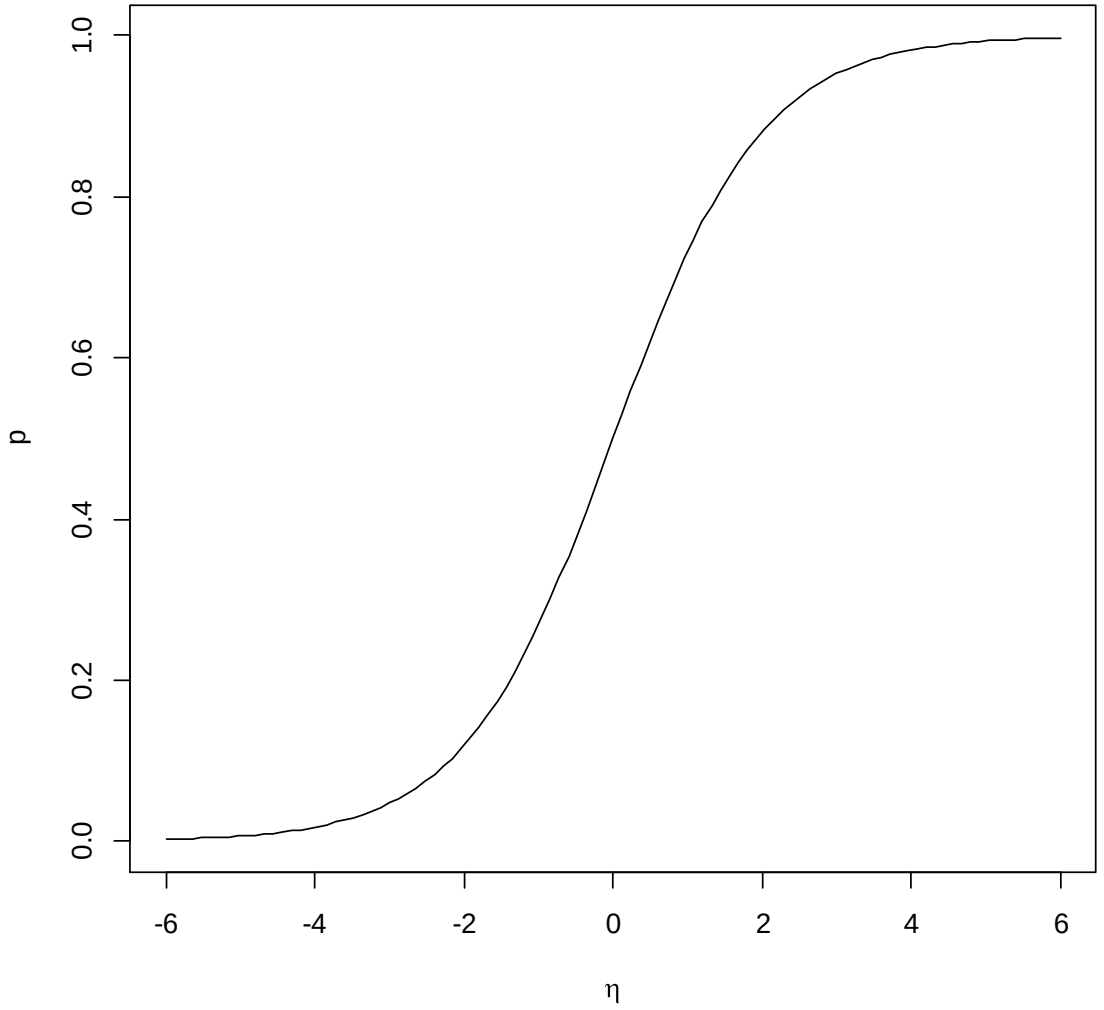
$$\text{logit}(p) = \log(\frac{p}{1-p}) = \beta_1 + \beta_2 X$$

# Is "logit" the only transformation that works?

$$\text{logit}(p) = \eta = \beta_1 + \beta_2 X$$

$$F^{-1}(p) = \eta = \beta_1 + \beta_2 X$$

$$p = F(\eta) = F(\beta_1 + \beta_2 X)$$



# Modeling a binary response using a general link function

- Suppose the binary response

$$Y \sim \text{Bernoulli}(p)$$

- We will make two additional assumptions on top of this assumption:
  1. Y relies on X only through its mean, e.g $Y \sim \text{Bernoulli}(p(X))$
  2. The transformation of the parameter $p$

$$F^{-1}(p) = \beta_1 + \beta_2 X \text{ or } p = F(\beta_1 + \beta_2 X)$$

$F$ could be **any cumulative distribution function**. We call $F^{-1}$ the **link function .**

# Examples of the link function

- Probit: $\eta = \Phi^{-1}(p)$. ⟺ Normal-distributed latent variables.
- Complementary log-log: $\eta = \log(-\log(1-p))$ ⟺ Gumbel-distributed latent variables.
- Cauchit: $\eta = \tan^{-1}(\pi(p-1/2))$ Cauchy-distributed latent variables.

# Compare the models with different link functions

```
lmod.0<-glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial,wcgs)

lmod.probit<-
glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial(link=probit),wcgs)

lmod.cloglog<-
glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial(link=cloglog),wcgs)

lmod.cauchit<-
glm(chd~age+height+bmi+dbp+chol+dibep+cigs,family=binomial(link=cauchit),wcgs)

round(coef(lmod.0),3)
round(coef(lmod.probit),3)
round(coef(lmod.cloglog),3)
round(coef(lmod.cauchit),3)
```

# Compare the coefficients

```
> round(coef(lmod.0),3)
(Intercept)         age      height         bmi         dbp        chol      dibepB        cigs
    -16.071       0.068       0.055       0.057       0.022       0.011       0.658       0.022

> round(coef(lmod.probit),3)
(Intercept)         age      height         bmi         dbp        chol      dibepB        cigs
     -8.608       0.034       0.030       0.033       0.012       0.006       0.339       0.012

> round(coef(lmod.cloglog),3)
(Intercept)         age      height         bmi         dbp        chol      dibepB        cigs
    -14.582       0.063       0.047       0.046       0.021       0.010       0.607       0.020

> round(coef(lmod.cauchit),3)
(Intercept)         age      height         bmi         dbp        chol      dibepB        cigs
    -24.923       0.144       0.051       0.015       0.052       0.018       1.640       0.038
```

# Compare the predicted values

```
predval<-sapply(list(lmod.0,lmod.probit,lmod.cloglog,lmod.cauchit),fitted)
colnames(predval)<-c("logit","probit","cloglog","cauchit")

round(predval[1:10,],3)

round(predval[fitted(lmod.0)>0.3 & fitted(lmod.0) <0.5,],3)

round(predval[fitted(lmod.0)<0.01,],3)

summary(fitted(lmod.0))
```

# The first 10 predictions

```
> round(predval[1:10,],3)
     logit probit cloglog cauchit
2001 0.100  0.102   0.100   0.098
2002 0.042  0.041   0.044   0.064
2003 0.013  0.009   0.015   0.042
2004 0.010  0.006   0.012   0.040
2005 0.143  0.143   0.141   0.117
2006 0.029  0.027   0.030   0.049
2007 0.014  0.010   0.016   0.043
2008 0.009  0.005   0.011   0.040
2009 0.022  0.018   0.023   0.045
2010 0.160  0.168   0.153   0.115
```

# Prediction in the midrange of the probability curve

```
> round(predval[fitted(lmod.0)>0.3 & fitted(lmod.0)
<0.5,],3)
      logit probit cloglog cauchit
2034  0.345  0.328   0.342   0.299
2140  0.413  0.384   0.419   0.423
2160  0.412  0.385   0.421   0.476
2252  0.340  0.336   0.330   0.225
2255  0.342  0.332   0.336   0.264
2280  0.394  0.384   0.382   0.223
2288  0.318  0.304   0.317   0.319
3143  0.387  0.361   0.393   0.435
3182  0.337  0.330   0.323   0.196
```

# Prediction on the tail of the probability curve

```
> round(predval[fitted(lmod.0)<0.01,],3)
      logit probit cloglog cauchit
2008  0.009  0.005   0.011   0.040
2050  0.009  0.005   0.011   0.039
2112  0.005  0.002   0.007   0.036
2155  0.009  0.006   0.011   0.038
3018  0.008  0.005   0.010   0.038
3030  0.008  0.005   0.009   0.035
3253  0.005  0.002   0.006   0.034
3459  0.008  0.004   0.009   0.037
3468  0.009  0.005   0.011   0.038
```

**Fig. 2** A graphical comparison of four link functions.