手把手教你用R语言评价临床预测模型，一文就够（附代码）

作者：阿琛 风间琉璃 来源：挑圈联靠

本文原地址：https://www.iikx.com/news/statistics/10597.html

手把手教你用R语言评价临床预测模型
。在日常的临床工作以及研究中，对于某个疾病，无论是肿瘤研究，还是非肿瘤研究，我们常听到患者提出这样的问题，“ 我的检查结果是这样的，那么最终患病的概率有多少，生存情况又是怎样的呢 ”。当然，生信分析，作为医学研究的三大主线之一，亦是如此，最终的结局无外乎两种，一是发病率是多少，二是预后生存情况如何。

（一） Nomogram图：

当我们通过数据挖掘，或者模型构建，发现了一种新的Biomarker，或者风险模型，除了通过ROC曲线或者生存分析评估其对疾病进展或者预后的独立预测能力以外，另一种很重要的手段就是该分子变量与其他已有的临床病理特征结合，综合预测患病率或生存率模型的重要能力。如果我们能提前预测病人病情的进展情况，那么有时候将会做出不同的临床决定，使整个过程更偏向个性化治疗。
Nomogram图，又称为列线图，是基于多因素分析的结果，将多个预测指标进行整合，根据一定的比例分配
，以图形的形式将各个
变量之间对结局预测之间相互关系进行可视化
展示。下面，一起学习一下基于Logistic回归和Cox回归分析的列线图的绘制过程。

基于Logistic回归的列线图：

1. 引用R包：

```
#install.packages("rms")

library(rms) #引用rms包
```

2. 读取文件：

```
setwd("C:Users00Desktop9_Nomogram") #设置工作目录

rt <- read.table("Log.txt",header=T,sep="") #读取数据
```

head(rt) #查看数据集rt

	id	Status	Age	Gender	BMI	Education	Alcohol
1	1	1	1	0	0	1	0
2	2	1	1	0	1	1	0
3	3	0	0	1	0	0	0
4	4	0	1	0	0	2	0
5	5	1	1	0	1	0	1
6	6	1	0	0	1	0	0

▲在该数据集中，主要包含了年龄(Age)，性别(Gender)，BMI值，教育水平(Education)，饮酒史(Alcohol)5个自变量，以及1个结局变量(Status)。

3. 设置变量参数：

首先，根据分组情况，对所有的变量添加标签

rt$Age <- factor(rt$Age,labels=c("<60",">=60"))

rt$Gender <- factor(rt$Gender,labels=c("No","Yes"))

rt$BMI <- factor(rt$BMI,labels=c("0","1","2"))

rt$Education <- factor(rt$Education,labels=c("Primary","Secondary","Higher"))

rt$Alcohol <- factor(rt$Alcohol,labels=c("No","Yes"))随后，使用datadist()函数将数据打包ddist <- datadist(rt) #使用函数datadist()将数据打包

options(datadist = "ddist")

4. 构建Logistic模型：

fit <- lrm(Status~Age + Gender + BMI + Education + Alcohol, data=rt, x=T, y=T)

#利用lrm()函数对模型进行拟合

fit #查看模型拟合结果

结果如下：

```
Logistic Regression Model

 lrm(formula = Status ~ Age + Gender + BMI + Education + Alcohol,
     data = rt, x = T, y = T)

                       Model Likelihood     Discrimination    Rank Discrim.
                          Ratio Test           Indexes           Indexes
 Obs           735    LR chi2     254.83    R2       0.413    C       0.839
  0            508    d.f.             7    g        1.558    Dxy     0.679
  1            227    Pr(> chi2) <0.0001    gr       4.752    gamma   0.723
 max |deriv| 2e-08                          gp       0.281    tau-a   0.290
                                            Brier    0.141

                     Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept          -2.2399 0.1921 -11.66 <0.0001
 Age=>=60           -0.0602 0.1989  -0.30 0.7622
 Gender=Yes          0.6412 0.2378   2.70 0.0070
 BMI=1               2.5794 0.2040  12.65 <0.0001
 BMI=2              -0.5508 1.1488  -0.48 0.6316
 Education=Secondary 1.4312 0.2970   4.82 <0.0001
 Education=Higher   -1.5375 1.1023  -1.39 0.1631
 Alcohol=Yes         1.0090 0.3076   3.28 0.0010
```

在模型中，纳入所有的变量构建了Logistic模型，结果展示了每个变量在模型中的系数以及P值，而且模型的C指数为0.839，展示出模型良好的预测能力。

5.构建Nomogram及可视化：

使用nomogram()函数构建模型

nom<- nomogram(fit, fun=plogis,

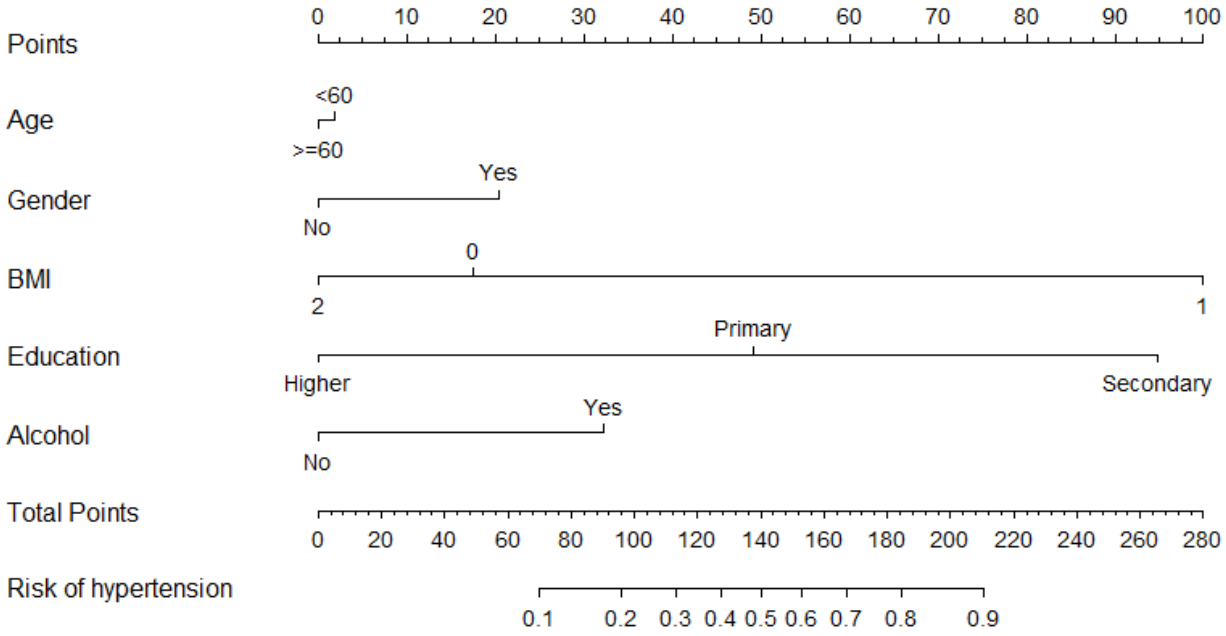
fun.at=c(0.0001,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.9999),

lp=F,

funlabel="Risk of hypertension") #构建Nomogram图

plot(nom) #输出Nomogram图

结果如下：



通过将每个指标对应的Points相加，与Total points相对应，即可获得不同患者的患病可能性，并且可以提前对高危的患者进行针对性的干预。

基于Cox回归的列线图：

与Logistic回归的列线图一致，首先便是数据的读取与整理过程。

1.引用R包：

library(rms)

library(foreign)

library(survival)

2.读取文件：

setwd("C:Users00Desktop9_Nomogram") #设置工作目录

rt2 <- read.table("Cox.txt",header=T,sep=" ") #读取数据

head(rt2) #查看数据集rt2

```
            Id    futime fustat gender  stage  T  M  N risk
1 TCGA-HU-8244 2.0328767      0      F Stage1 T1 M0 N0  low
2 TCGA-HU-A4G9 2.0164384      0      F Stage1 T1 M0 N0  low
3 TCGA-D7-6528 1.2684932      0      F Stage1 T2 M0 N0  low
4 TCGA-RD-A8N2 9.6986301      0      F Stage1 T2 M0 N0 high
5 TCGA-RD-A7BW 0.4273973      1      F Stage1 T2 M0 N0 high
6 TCGA-BR-7707 1.0547945      0      F Stage1 T2 M0 N0  low
```

3.设置变量参数：

```
rt2$gender <- factor(rt2$gender,labels=c("F", "M"))

rt2$stage <- factor(rt2$stage,labels=c("Stage1", "Stage2", "Stage3", "Stage4"))

rt2$T <- factor(rt2$T,labels=c("T1", "T2", "T3", "T4"))

rt2$M <- factor(rt2$M,labels=c("M0", "M1"))

rt2$N <- factor(rt2$N,labels=c("N0", "N1", "N2", "N3"))

rt2$risk <- factor(rt2$risk,labels=c("low", "high"))

ddist <- datadist(rt2) #使用函数datadist()将数据打包

options(datadist='ddist')
```

4.构建Cox回归模型：

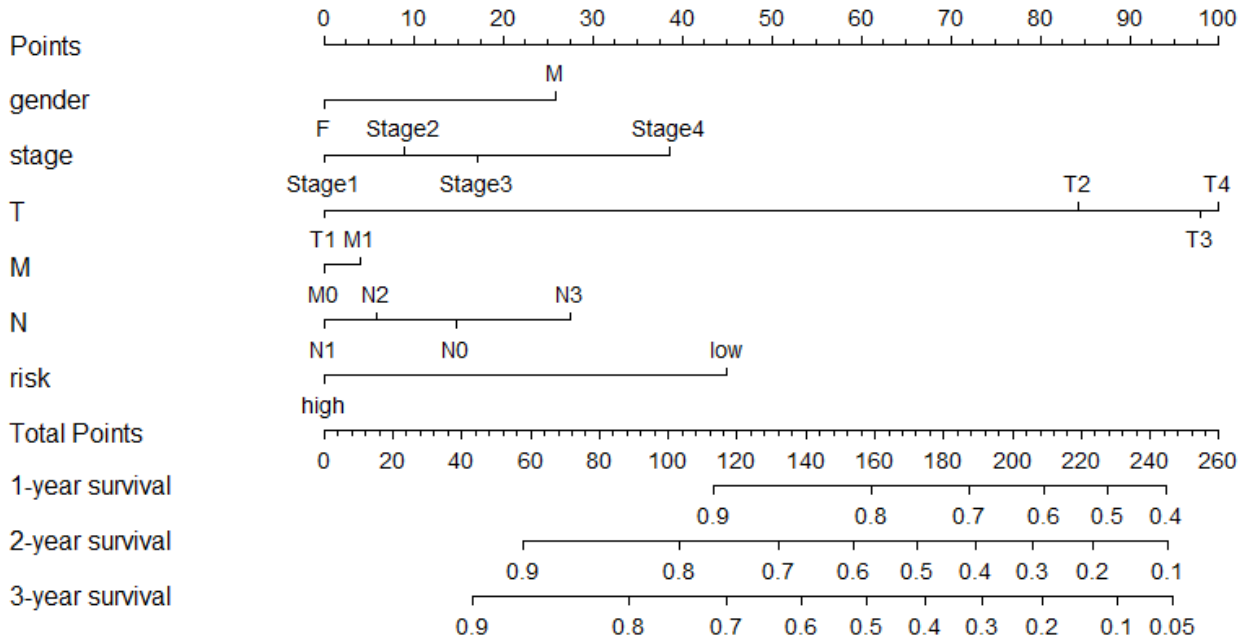
```
f <- cph(Surv(futime, fustat) ~gender + stage +T + M + N + risk,

x=T, y=T, surv=T,

data=rt2, time.inc=1)

surv <- Survival(f)
```

```
Cox Proportional Hazards Model

 cph(formula = Surv(futime, fustat) ~ gender + stage + T + M +
     N + risk, data = rt2, x = T, y = T, surv = T, time.inc = 1)

                      Model Tests       Discrimination
                                           Indexes
 Obs        296    LR chi2     34.77    R2        0.114
 Events     107    d.f.           12    Dxy       0.329
 Center 1.5752    Pr(> chi2) 0.0005    g         0.694
                  Score chi2  35.33    gr        2.002
                  Pr(> chi2) 0.0004

               Coef    S.E.   Wald Z Pr(>|Z|)
 gender=M      0.4270 0.2240  1.91   0.0567
 stage=Stage2  0.1466 0.5466  0.27   0.7886
 stage=Stage3  0.2823 0.7409  0.38   0.7032
 stage=Stage4  0.6356 0.7611  0.84   0.4036
 T=T2          1.3870 1.0556  1.31   0.1888
 T=T3          1.6114 1.1150  1.45   0.1484
 T=T4          1.6462 1.1313  1.46   0.1456
 M=M1          0.0654 0.4912  0.13   0.8940
 N=N1         -0.2443 0.3839 -0.64   0.5246
 N=N2         -0.1470 0.4721 -0.31   0.7556
 N=N3          0.2076 0.4760  0.44   0.6627
 risk=high    -0.7402 0.2067 -3.58   0.0003
```

5.构建Nomogram及可视化：

```
nom2 <- nomogram(f,

fun=list(function(x) surv(1, x), function(x) surv(2, x), function(x) surv(3, x)),

lp=F,

funlabel=c("1-year survival", "2-year survival", "3-year survival"),

maxscale=100,fun.at=c(0.99, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3,0.2,0.1,0.05)) #构建Nomogram图

plot(nom2) #输出Nomogram图
```

结果如下：



在模型中，共纳入了6个变量，通过整合，预测患者1年，2年以及3年的预后生存情况。

Nomogram图的绘制就到此结束了，又点亮了一颗技能之星

（二）临床模型评价：

当临床预测模型成功建立，就会面临一个问题，如何去评价模型的质量。模型的评价一般是通过比较人群的模型预测结果与实际观测结果，来评价预测模型的效果。

对于模型的评价指标，我们往往分成三种：

1. 区分度评价指标：C指数(C-Index)，重新分类指数(Net reclassification index，NRI);

2. 一致性评价指标：校正曲线(Calibration plot);

3. 临床有效性评价指标：决策分析曲线(Decision Curve Analysis， DCA);

今天就让阿琛来给大家分别讲讲如何评价前期建立的临床预测模型。

1.建立预测模型：

首先，我们一起快速的回顾一下如何构建Cox回归模型。

1.1 引用R包：

#install.packages("rms")

#install.packages("foreign")

#install.packages("survival")

library(rms)

library(foreign)

library(survival)

1.2 读取文件：

setwd("C:Users00Desktop13_clinical") #设置工作目录

rt <- read.table("clinical.txt",header=T,sep="") #读取数据

head(rt) #查看前5行的数据

	Id	futime	fustat	gender	stage	T	M	N	risk
1	TCGA-HU-8244	2.0328767	0	F	Stage1	T1	M0	N0	low
2	TCGA-HU-A4G9	2.0164384	0	F	Stage1	T1	M0	N0	low
3	TCGA-D7-6528	1.2684932	0	F	Stage1	T2	M0	N0	low
4	TCGA-RD-A8N2	9.6986301	0	F	Stage1	T2	M0	N0	high
5	TCGA-RD-A7BW	0.4273973	1	F	Stage1	T2	M0	N0	high
6	TCGA-BR-7707	1.0547945	0	F	Stage1	T2	M0	N0	low

在该数据集中，共包含6个不同的临床病理特征以及患者的生存状态和生存时间。

1.3 构建Cox回归模型：

首先，对数据的相关参数进行整理与设置：

rt$gender <- factor(rt$gender,labels=c("F", "M"))

rt$stage <- factor(rt$stage,labels=c("Stage1", "Stage2", "Stage3", "Stage4"))

rt$T <- factor(rt$T,labels=c("T1", "T2", "T3", "T4"))

rt$M <- factor(rt$M,labels=c("M0", "M1"))

rt$N <- factor(rt$N,labels=c("N0", "N1", "N2", "N3"))

rt$risk <- factor(rt$risk,labels=c("low", "high"))

str(rt)

ddist <- datadist(rt) #将数据打包

options(datadist='ddist')

units(rt$futime) <- "year" #定义时间的单位

```
> str(rt)
'data.frame':   296 obs. of  9 variables:
 $ Id    : chr  "TCGA-HU-8244" "TCGA-HU-A4G9" "TCGA-D7-6528" "TCGA-RD-A8N2" ...
 $ futime: num  2.033 2.016 1.268 9.699 0.427 ...
 $ fustat: int  0 0 0 0 1 0 0 0 0 0 ...
 $ gender: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ stage : Factor w/ 4 levels "Stage1","Stage2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ T     : Factor w/ 4 levels "T1","T2","T3",..: 1 1 2 2 2 2 2 2 2 2 ...
 $ M     : Factor w/ 2 levels "M0","M1": 1 1 1 1 1 1 1 1 1 1 ...
 $ N     : Factor w/ 4 levels "N0","N1","N2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ risk  : Factor w/ 2 levels "low","high": 2 2 2 1 1 2 1 2 2 2 ...
```

接着，将所有变量纳入模型中，构建Cox回归模型：

#构建回归模型

f <- cph(Surv(futime, fustat) ~gender + stage +T + M + N + risk,

x=T, y=T, surv=T, data=rt)

f #查看模型的相关参数

```
Cox Proportional Hazards Model

 cph(formula = Surv(futime, fustat) ~ gender + stage + T + M +
     N + risk, data = rt, x = T, y = T, surv = T)

                       Model Tests    Discrimination
                                            Indexes
 Obs        296    LR chi2     34.77    R2        0.114
 Events     107    d.f.           12    Dxy       0.329
 Center 1.5752    Pr(> chi2) 0.0005    g         0.694
                  Score chi2  35.33    gr        2.002
                  Pr(> chi2) 0.0004

               Coef    S.E.   Wald Z Pr(>|Z|)
 gender=M      0.4270 0.2240  1.91   0.0567
 stage=Stage2  0.1466 0.5466  0.27   0.7886
 stage=Stage3  0.2823 0.7409  0.38   0.7032
 stage=Stage4  0.6356 0.7611  0.84   0.4036
 T=T2          1.3870 1.0556  1.31   0.1888
 T=T3          1.6114 1.1150  1.45   0.1484
 T=T4          1.6462 1.1313  1.46   0.1456
 M=M1          0.0654 0.4912  0.13   0.8940
 N=N1         -0.2443 0.3839 -0.64   0.5246
 N=N2         -0.1470 0.4721 -0.31   0.7556
 N=N3          0.2076 0.4760  0.44   0.6627
 risk=high    -0.7402 0.2067 -3.58   0.0003
```

自此，我们的Cox回归模型就构建完成了。接下来就是从不同的角度来评价模型的质量。

2.C指数(C-index)的计算：

validate(f, method="boot", B=1000, dxy=T) #重复模拟1000次

rcorrcens(Surv(futime, fustat) ~ predict(f), data = rt)

结果显示：

```
Somers' Rank Correlation for Censored Data    Response variable:Surv(futime, fustat)

               C    Dxy  aDxy    SD    Z P   n
predict(f) 0.336 -0.329 0.329 0.056 5.86 0 296
```

在此模型中，C-index为0.664，即1-0.336=0.664。

当然，根据C指数而言，该模型的预测区分度并不是太高。一般而言，我们将0.51-0.7认为是低可信度，0.71-0.9为中等可信度，> 0.9为高可信度。

3.Calibration plot的绘制：

绘制三年生存率的校准曲线

#计算三年生存率的校准曲线

f3 <- cph(Surv(futime, fustat) ~gender + stage +T + M + N + risk,

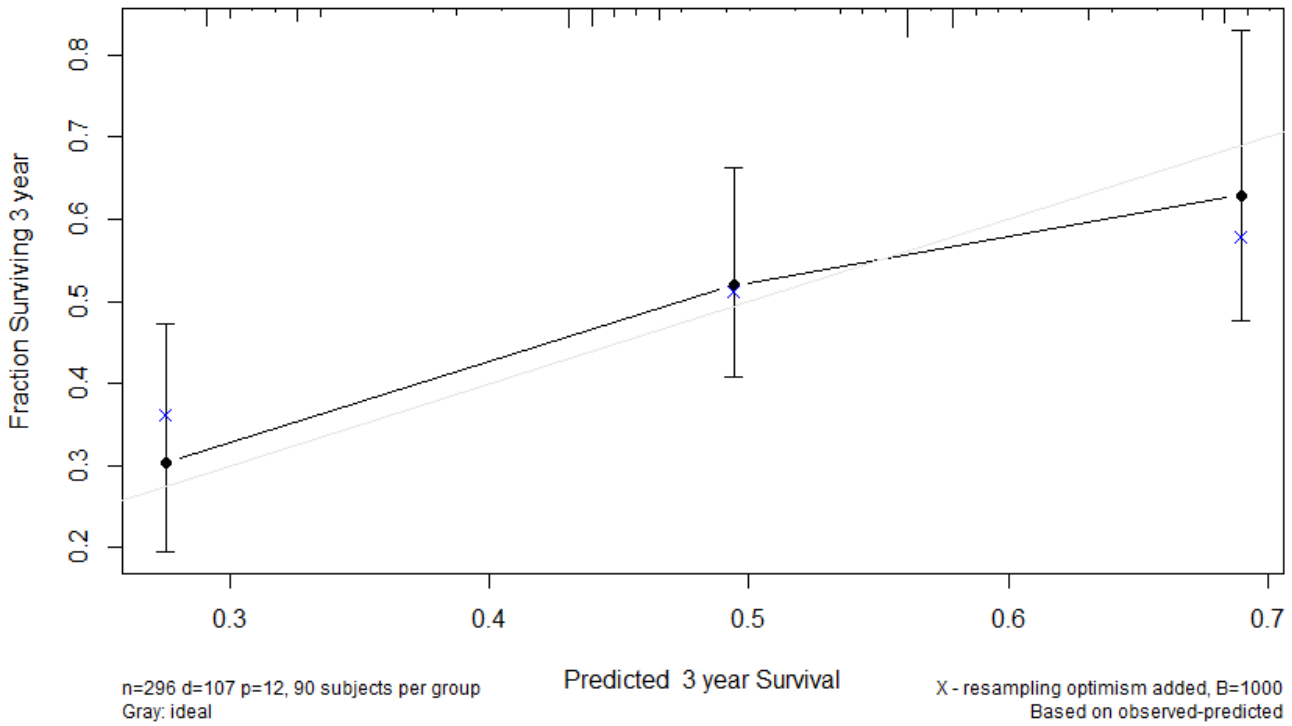
x=T, y=T, surv=T, time.inc = 3, data=rt)

cal3 <- calibrate(f3, cmethod="KM", method="boot", u= 3, m= 90, B= 1000)

#一般将患者分成3组，即m约等于患者总数/3，且重复模拟1000次

plot(cal3)

结果显示：



绘制五年生存率的校准曲线

#计算五年生存率的校准曲线

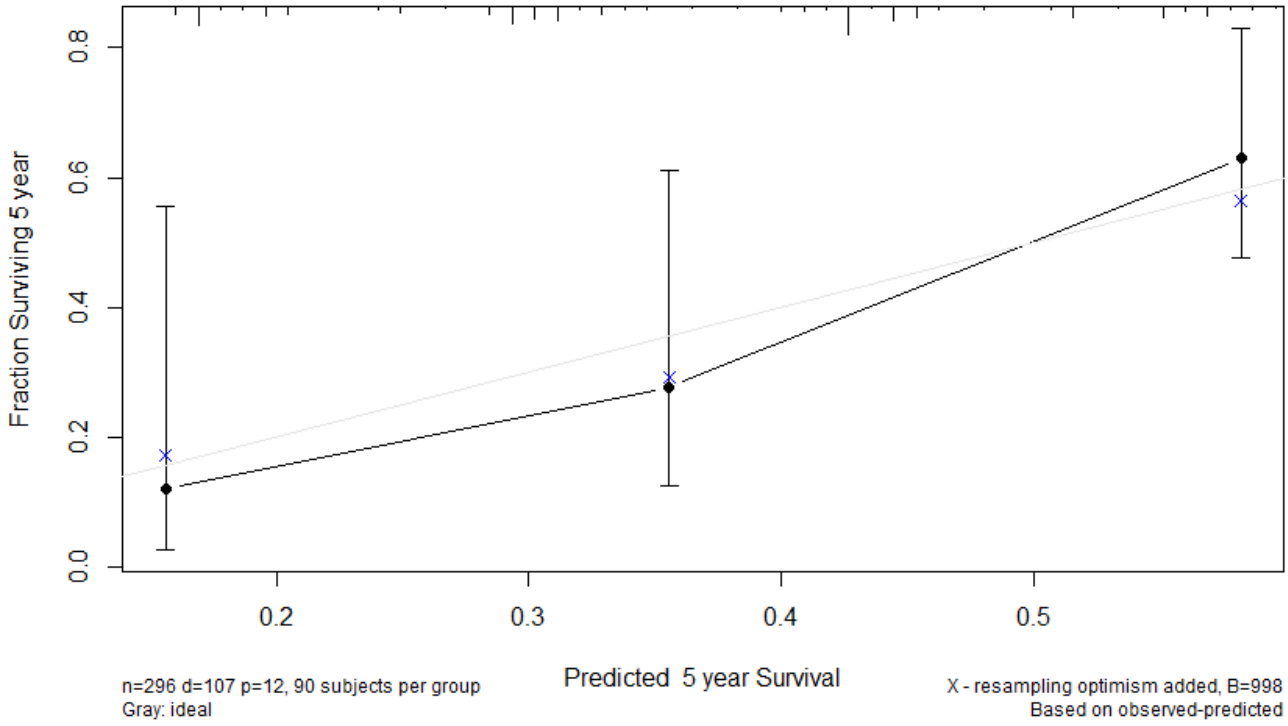
f5 <- cph(Surv(futime, fustat) ~gender + stage +T + M + N + risk,

x=T, y=T, surv=T, time.inc = 5, data=rt)

cal5 <- calibrate(f5, cmethod="KM", method="boot", u= 5, m= 90, B= 1000)

plot(cal5)

结果显示：



对于Calibration plot，横坐标为模型预测得到的患者生存率，纵坐标为患者实际观察得到的生存率，三点之间的联系与虚线之间越相近，越能说明模型良好的一致性。

4.DCA曲线的绘制：

source("stdca.R") #把stdca.R放在之前设定的工作目录中，并进行引用

rt$three.years.Survival.Probabilitynew = c(1- (summary(survfit(f, newdata = rt),times = 3)$surv))
#计算3年生存率

rt$five.years.Survival.Probabilitynew = c(1- (summary(survfit(f, newdata = rt),times = 5)$surv))
#计算5年生存率

write.csv(rt, "DCA.Survival.csv")

#将患者的3年和5年生存率结果保存

随后，根据患者的生存情况，分别绘制3年和5年的DCA曲线;

#绘制3年生存率的DCA曲线

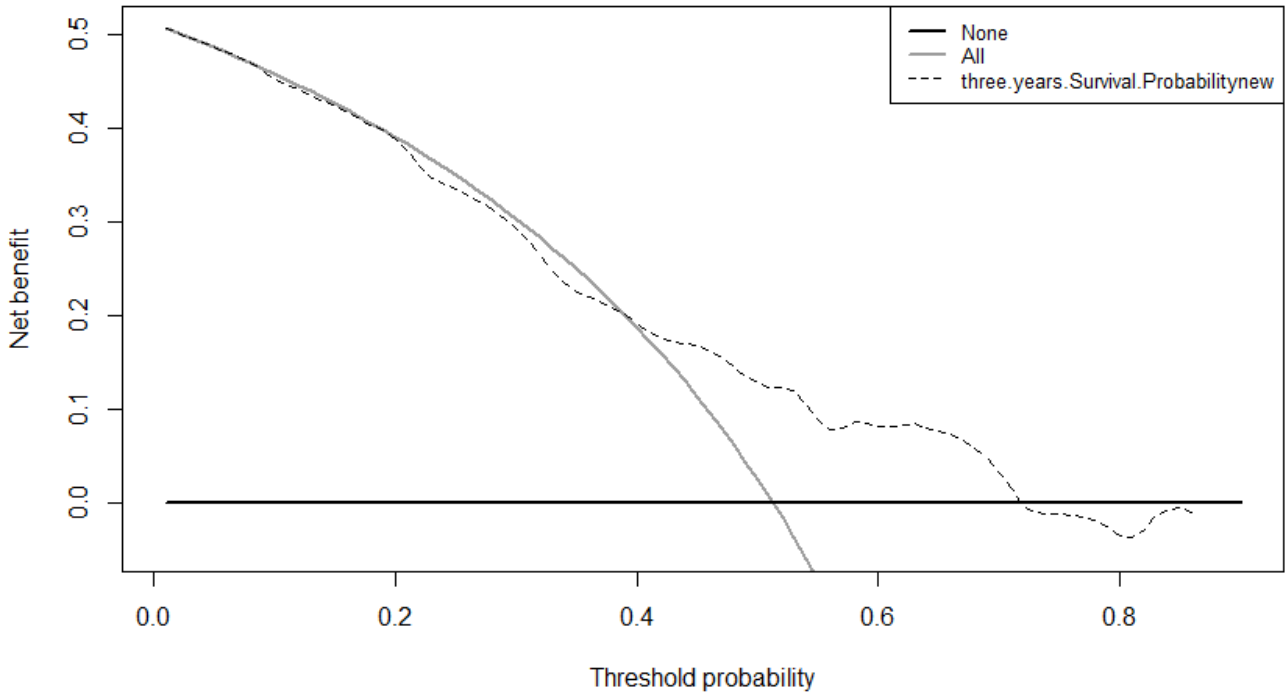
stdca(data=rt,

outcome="fustat", ttoutcome="futime", timepoint=3,

predictors="three.years.Survival.Probabilitynew",

xstop=0.9, smooth=TRUE)

结果显示：



#绘制5年生存率的DCA曲线

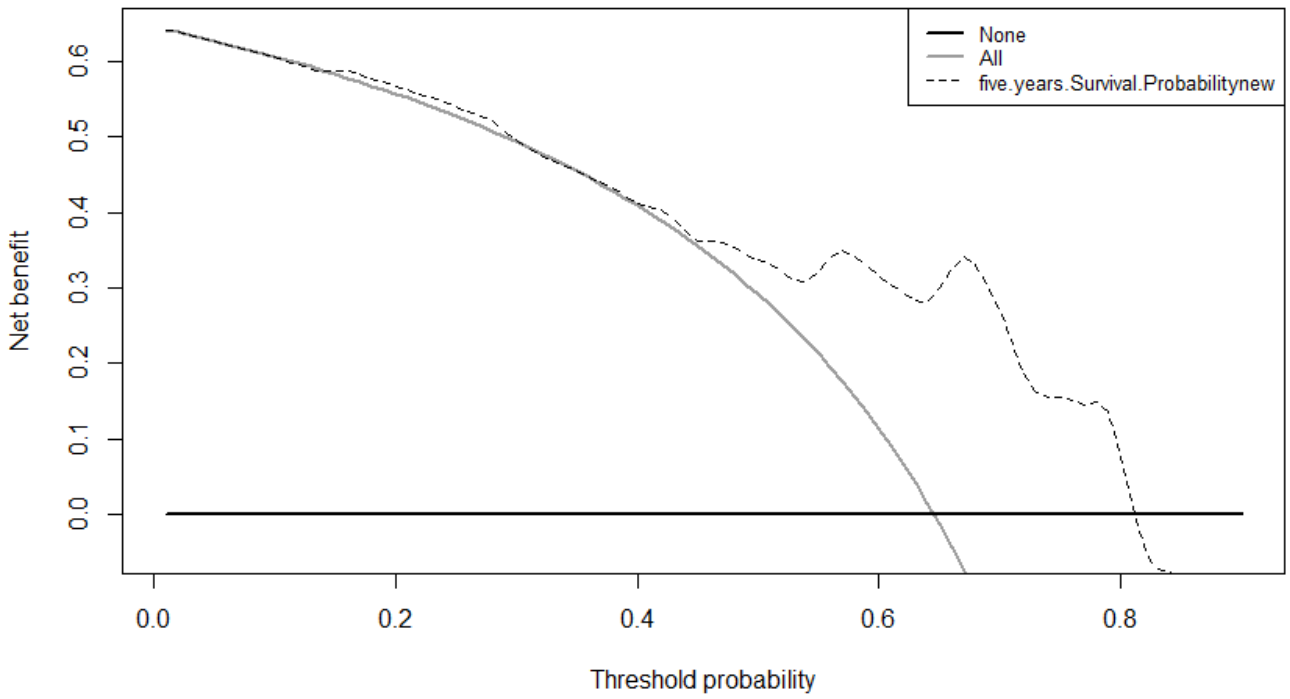
stdca(data=rt,

outcome="fustat", ttoutcome="futime", timepoint=5,

predictors="five.years.Survival.Probabilitynew",

xstop=0.9,smooth=TRUE)

结果显示：



5.NRI指数的计算：

对于NRI指数的计算，有2个专门的R包，分别为nricens和PredictABEL;其中，nricens计算出来的是绝对的NRI，PredictABEL则为相对的NRI，一般在使用过程中选择其中之一即可。

#install.packages("nricens")

library(nricens)

随后，分别构建两个不同的模型，以进行比较：

mstd <- cph(Surv(futime, fustat) ~gender + stage +T + M + N,

x=T, y=T, surv=T, data=rt) #使用除risk外的变量进行建模

mnew <- cph(Surv(futime, fustat) ~gender + stage +T + M + N + risk,

x=T, y=T, surv=T, data=rt) #该模型包含所有的变量

接着，计算NRI指数;

#计算连续的NRI，取值为5%时就定义为重分类

#3年生存率NRI

nricens(mdl.std = mstd, mdl.new = mnew, t0 = 3, updown = 'diff',cut = 0.05, niter = 200)

#5年生存率NRI

nricens(mdl.std = mstd, mdl.new = mnew, t0 = 5, updown = 'diff',cut = 0.05, niter = 200)

#计算分类变量计算的NRI

nricens(mdl.std = mstd, mdl.new = mnew, t0 = 5, updown = 'category',cut = c(0.3,0.6), niter = 200)

对于NRI的计算，其主要是比较新旧两个模型之间存在区分度。其中，updown主要定义了一个样本的风险是否变动的方式，category是指分类值，即熟悉的低、中、高风险，另有一种diff，为连续值。

Cut是判断风险高低的临界值。当updown为diff时，cut只需设置1个值，比如0.05，即认为当预测的风险在新旧模型中相差5%时，即被认为是重新分类了;而当updown为category时，可以对cut设置两个不同的值，即0~29%为低风险，30%~59%为中风险，60%~100%为高风险。

结果显示：

```
Point estimates:
                 Estimate
NRI             0.2616104
NRI+            0.0903475
NRI-            0.1712629
Pr(Up|Case)     0.5265889
Pr(Down|Case)   0.4362414
Pr(Down|Ctrl)   0.5462602
Pr(Up|Ctrl)     0.3749973

Point estimates:
                 Estimate
NRI             0.5257705
NRI+            0.1458935
NRI-            0.3798771
Pr(Up|Case)     0.5697149
Pr(Down|Case)   0.4238214
Pr(Down|Ctrl)   0.6100733
Pr(Up|Ctrl)     0.2301963
```

当updown为diff时，3年和5年的NRI值分别为0.26和0.53，均大于5%，可以认为预测的风险在新旧模型中发生了重新分类;

3年生存率NRI的95%可信区间：

```
Now in bootstrap..

Point & Interval estimates:
               Estimate       Lower      Upper
NRI           0.5257705 -0.05381771 1.0123101
NRI+          0.1458935 -0.09386830 0.3505087
NRI-          0.3798771 -0.05306022 0.7201348
Pr(Up|Case)   0.5697149  0.21856424 0.6866345
Pr(Down|Case) 0.4238214  0.10098861 0.5094708
Pr(Down|Ctrl) 0.6100733  0.10825637 0.8395900
Pr(Up|Ctrl)   0.2301963  0.00000000 0.3764385
```

5年生存率NRI的95%可信区间：

```
Now in bootstrap..

Point & Interval estimates:
                 Estimate       Lower      Upper
NRI            0.10819237 -0.1652456 0.5293685
NRI+          -0.04947369 -0.1999658 0.2319574
NRI-           0.15766606 -0.1362963 0.4742668
Pr(Up|Case)    0.16696376  0.0000000 0.3387372
Pr(Down|Case)  0.21643745  0.0000000 0.2884014
Pr(Down|Ctrl)  0.27297859  0.0000000 0.5250842
Pr(Up|Ctrl)    0.11531253  0.0000000 0.2675773
```

同时，由于做了200次重复取样，相当于有200个NRI，于是NRI就有了标准误和95%的置信区间。同理，可以得到当updown为category时的NRI值。

到此，对于临床预测模型评价指标的介绍就到此结束了。大家可以对照着整个分析过程，使用示例数据或者自己的模型数据，进行相应的学习~

（三）Logistic回归模型评价：

在常用的临床模型构建中，主要分为两种，包括临床预测模型(Cox回归模型)和临床诊断模型(Logstic回归模型)。在之前的内容中，阿琛给大家介绍了如何使用Nomogram图将临床预测模型可视化，以及Cox回归模型的相关评价指标。

下面是临床模型的第三集临床诊断模型篇，即如何对Logistic回归模型进行评价。

1.建立Logistic预测模型：

1.1 引用R包：

```
#install.packages("foreign")

#install.packages("rms")

#install.packages("pROC")
```

#install.packages("rmda")

#install.packages("nricens")

library(foreign)

library(rms) #构建Logstic模型

library(pROC) #绘制ROC曲线

library(rmda) #绘制DCA曲线

library(nricens) #计算NRI值

1.2 读取文件：

setwd("C:Users00Desktop14_Logstic") #设置工作目录

rt <- read.table("clinical_2.txt",header=T,sep=" ") #读取数据

head(rt) #查看前5行的数据

可以发现，该数据集主要包括5个不同的变量和1个结局变量;在此，我们主要通过构建临床诊断模型，来预测结局Status。

	id	Status	Age	Gender	BMI	Education	Alcohol
1	1	1	1	0	19.53213	1	0
2	2	1	1	0	16.39798	1	0
3	3	1	0	1	18.93878	0	0
4	4	1	1	0	19.41176	2	0
5	5	1	1	0	21.95312	0	1
6	6	1	0	0	21.96712	0	0

1.3 构建Logstic模型：

首先，对数据集中的相关参数进行设置;

rt$Age <- factor(rt$Age,labels=c("<60",">=60"))

rt$Gender <- factor(rt$Gender,labels=c("No","Yes"))

rt$Education <- factor(rt$Education,labels=c("Primary","Secondary","Higher"))

rt$Alcohol <- factor(rt$Alcohol,labels=c("No","Yes"))

str(rt)

ddist <- datadist(rt)

options(datadist = "ddist") #使用函数datadist()将数据打包

随后，使用glm()函数构建Logstic模型，在R中glm()函数可以拟合广义线性模型，其中包括Logstic回归模型。其中，我们必须使用family=binomial这个参数来告诉R运行Logstic回归模型，而不是其他的广义线性模型;

fit <- glm(Status~Age + Gender + BMI + Education + Alcohol, data=rt,

family = binomial(link="logit"), x=T)

#利用lrm()函数对模型进行拟合

summary(fit) #查看模型拟合结果

通过summary()函数，我们可以查看各个预测变量的系数及其P值;可以看到，只有两个特征的P值小于0.05(BMI和Education);

```
Call:
glm(formula = Status ~ Age + Gender + BMI + Education + Alcohol,
    family = binomial(link = "logit"), data = rt, x = T)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8828  -0.9301  -0.6179   1.0182   2.5097

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        4.35854    0.97365   4.476 7.59e-06 ***
Age>=60            0.09774    0.26595   0.368  0.71324
GenderYes          0.33493    0.32554   1.029  0.30355
BMI               -0.22396    0.04129  -5.424 5.84e-08 ***
EducationSecondary 0.62087    0.41804   1.485  0.13749
EducationHigher    2.13577    0.75316   2.836  0.00457 **
AlcoholYes        -0.06455    0.42279  -0.153  0.87866
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 415.72  on 313  degrees of freedom
Residual deviance: 361.42  on 307  degrees of freedom
AIC: 375.42

Number of Fisher Scoring iterations: 4
```

2.计算C指数及绘制ROC曲线：

2.1 C-Index的计算：

Cindex <- rcorrcens(Status~predict(fit), data = rt)

Cindex

通过这种方式，我们也可以获得该模型的C-Index为0.718，标准差为0.061;一般而言，我们将<0.5看成模型没有任何预测能力，0.51-0.7认为是较差的准确性，0.71-0.9为中等的准确性，>0.9为高度的准确性。

```
Somers' Rank Correlation for Censored Data    Response variable:Status

                C   Dxy  aDxy    SD    Z P   n
predict(fit) 0.718 0.436 0.436 0.061 7.15 0 314
```

2.2 ROC曲线的绘制：

接下来，加个餐，我们一起来看下如何绘制预测模型的ROC曲线;

gfit <- roc(Status~predict(fit), data = rt)

plot(gfit,

print.auc=TRUE, #输出AUC值

print.thres=TRUE, #输出cut-off值

main = "ROC CURVE", #设置图形的标题

col= "red", #曲线颜色

print.thres.col="black", #cut-off值字体的颜色

identity.col="blue", #对角线颜色

identity.lty=1,identity.lwd=1)

结果显示

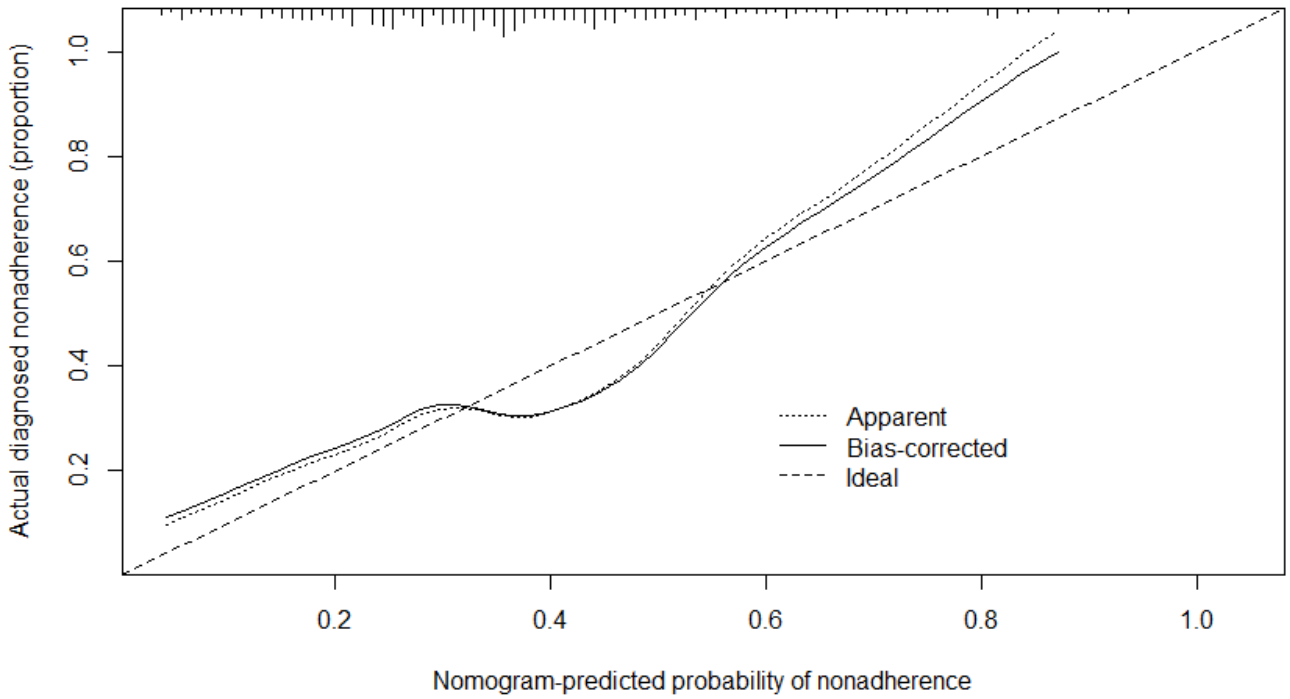
该模型ROC曲线的曲线下面积(AUC值)为0.718;对于AUC值，其含义与C-Index较为类似，我们往往将AUC位于0.5~0.7时定义为模型的效果较低，位于0.7~0.85之间为效果一般，而位于0.85~0.95时效果很好。同时，该模型的最佳截断值(即cut-off值)为0.011，也就是说当以0.011进行分组时，两组之间具有最佳的区分度。

3.基于ggstatspot包的绘制：

```
cal <- calibrate(fit, method="boot", B=1000)

plot(cal,

xlab="Nomogram-predicted probability of nonadherence",

ylab="Actual diagnosed nonadherence (proportion)",

sub=F)
```

如图所示

X轴为模型预测得到的结局可能性，而Y轴为实际观察得到的值，并重复计算1000次，其中Bias-corrected为校正曲线，而对角线Ideal为理想的曲线。校正曲线与理想曲线之间越相近，说明模型的预测能力越好。在该结果中，模型具有良好的校准能力。



4.绘制DCA曲线：

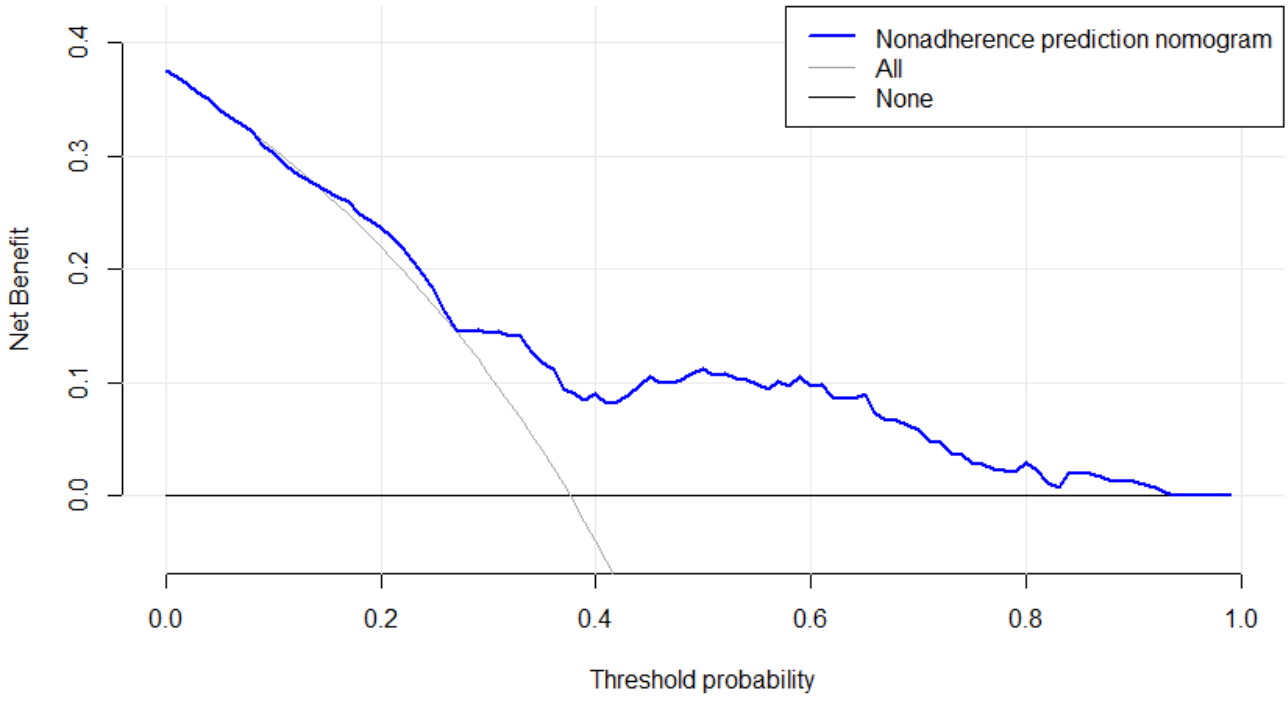
评估完模型的准确性，同时还需要进一步评估模型的获益率。因此，我们通过DCA曲线，来展示患者的净获益率。

```
modul<- decision_curve(data= rt,

Status~Age + Gender + BMI + Education + Alcohol,

family = binomial(link ='logit'),

thresholds= seq(0,1, by = 0.01),

confidence.intervals = 0.95)

plot_decision_curve(modul,

curve.names="Nonadherence prediction nomogram", #曲线名称

xlab="Threshold probability", #x轴名称

cost.benefit.axis =FALSE, col= "blue",

confidence.intervals=FALSE,
```

standardize = FALSE)

如图所示

蓝色的曲线为模型预测的获益情况，灰色的曲线为所有患者都接受干预的获益率，而横线为所有病人都不接受干预的获益率。取蓝色曲线与All的交点为起点，与None的交点为结束，在此范围内所对应的患者可以获益。



5.计算NRI指数：

在此，我们对比一下BMI的纳入对模型的质量是否会产生影响。首先，构建两个不同的预测模型;

##构建模型

fit_A <- glm(Status~Age + Gender + Education + Alcohol, data = rt, family = binomial(link="logit"),x=TRUE)

fit_B <- glm(Status~Age + Gender + Education + Alcohol + BMI, data = rt, family = binomial(link="logit"),x=TRUE)

接着，我们分别使用了两种不同的方式，进一步对两个模型进行了比较;

#计算连续的NRI，取值为5%时就定义为重分类

NRI <- nribin(mdl.std = fit_A, mdl.new = fit_B,

updown = 'diff',

cut = 0.05, niter = 500, alpha = 0.05)

结果显示

两个模型之间比较的NRI值为0.419(95%CI：0.247-0.671)，可以认为预测的风险在新旧模型中发生了重新分类;

```
Point estimates:
                Estimate
NRI            0.4190591
NRI+           0.2966102
NRI-           0.1224490
Pr(Up|Case)    0.5254237
Pr(Down|Case) 0.2288136
Pr(Down|Ctrl) 0.4489796
Pr(Up|Ctrl)    0.3265306

Now in bootstrap..

Point & Interval estimates:
                Estimate  Std.Error       Lower     Upper
NRI            0.4190591 0.10623805 0.24670719 0.6707904
NRI+           0.2966102 0.06683073 0.16346154 0.4260870
NRI-           0.1224490 0.05537590 0.05445545 0.2684211
Pr(Up|Case)    0.5254237 0.05850422 0.38738739 0.6229508
Pr(Down|Case) 0.2288136 0.03157810 0.16513761 0.2894737
Pr(Down|Ctrl) 0.4489796 0.05046880 0.35148515 0.5468750
Pr(Up|Ctrl)    0.3265306 0.02586536 0.24731183 0.3453608
```

#计算分类变量计算的NRI,只有概率超过0.011就定义为重分组了

#0.011是根据之前模型C的ROC分析确定的切点

NRI <- nribin(mdl.std = fit_A, mdl.new = fit_B,

updown = 'category',

cut = 0.011, niter = 500, alpha = 0.05)

结果显示

两个模型之间比较的NRI值为0.005(95%CI：0-0.021)，可以认为预测的风险在新旧模型中并没有发生重新分类;

```
Point estimates:
                 Estimate
NRI           0.005102041
NRI+          0.000000000
NRI-          0.005102041
Pr(Up|Case)   0.000000000
Pr(Down|Case) 0.000000000
Pr(Down|Ctrl) 0.005102041
Pr(Up|Ctrl)   0.000000000

Now in bootstrap..

Point & Interval estimates:
                 Estimate    Std.Error Lower      Upper
NRI           0.005102041 0.0066023136     0 0.02094241
NRI+          0.000000000 0.0004028951     0 0.00000000
NRI-          0.005102041 0.0067340860     0 0.02094241
Pr(Up|Case)   0.000000000 0.0000000000     0 0.00000000
Pr(Down|Case) 0.000000000 0.0004028951     0 0.00000000
Pr(Down|Ctrl) 0.005102041 0.0067340860     0 0.02094241
Pr(Up|Ctrl)   0.000000000 0.0000000000     0 0.00000000
```

好了，今天的分享就到此结束了。大家可以使用示例数据或者自己的模型数据，进行相应的学习。

（四）实例解析：

数据介绍：

首先我为大家介绍一下本次使用的数据.这次我选择的疾病是高血压数据。我们的数据一共有ID、Gender、Age、A、hypertension五个变量。该数据的目的是通过对每一个观测(每一行)的性别、年龄和指标A(这个指标可以是分子标志物、临床上的任意指标)，以及是否发生高血压疾病进行收集。构建预测模型。明确哪一类人是我们的潜在高血压疾病患者。从而将更多的健康宣传、金钱投入向这一类人群进行更多倾斜。

那么这儿有一个问题，那就是这个模型是否能够准确推断出我们预测的那一类人就是真正会得高血压的患者呢?这个问题推广一下，到医学领域也就是，我们构建的临床模型推测的潜在患病人群是否真的会患病呢?这就是模型的评价。那我们开始吧。

数据预处理：

第一步，我们需要导入数据

##setwd(“ D:/ROC and Calibration ”)

##并且不要ID这一列

dataset=read.csv('disease.csv')

dataset=dataset[2:5]

第二步，在构建模型之前需要处理数据

##将结局变量转换为因子型变量 ##将数据随机分为训练集和测试集，训练集的作用就是构建模型，测试集的目的就是构建的模型进行评价，我们采取7：3的比例随机将数据进行拆分。###install.packages(“ CaTools ”)

dataset$hypertension = factor(dataset$hypertension, levels = c(0, 1))

library(caTools)

set.seed(123)

split = sample.split(dataset$hypertension, SplitRatio = 0.7)

training_set = subset(dataset, split == TRUE)

test_set = subset(dataset, split == FALSE)

我们来看一看新的数据集的结构。

dim(training_set)

[1] 280 4

str(training_set)

'data.frame': 280 obs. of 4 variables:

$ Gender : Factor w/ 2 levels "Female","Male": 2 1 2 1 1 1 1 2 2 2 ...

$ Age : int 19 26 27 27 32 35 26 20 18 29 ...

$ A : int 19000 43000 58000 84000 150000 65000 80000 86000 82000 80000 ...

$ hypertension: Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...

好了，现在就得到我们想要的数据模型了。

构建模型，对logistic回归分析

有两个函数可以构建模型，分别是R本身的stats包里面的glm函数和rms包的lrm函数。我们这儿是使用的glm函数。

classifier = glm(formula = hypertension ~ .,

data = training_set,family ="binomial")

```
summary(classifier)
##
## Call:
## glm(formula = hypertension ~ ., family = "binomial", data = training_set)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -3.0629 -0.4945 -0.1071 0.2942 2.5220
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.383e+01 1.811e+00 -7.638 2.20e-14 ***
## GenderMale 2.660e-01 3.792e-01 0.701 0.483
## Age 2.599e-01 3.527e-02 7.368 1.73e-13 ***
## A 3.833e-05 6.833e-06 5.610 2.03e-08 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 364.98 on 279 degrees of freedom
## Residual deviance: 181.48 on 276 degrees of freedom
## AIC: 189.48
```

##

Number of Fisher Scoring iterations: 6

我们发现除了性别，年龄和薪资都是结局变量的重要预测因素(P<0.005)。

（五）模型的评价：

接下啦我们即将开始对模型进行评价，需要注意的是，我们对模型的评价应该要做三方面，分别是区分度(discrimination)、校正度(Calibration) 和决策曲线分析(decision curve analysis， DCA)。具体的理论知识我简单介绍一下。区分度就是模型能不能区分是或者否发生某个结局时间。

我们通常以ROC曲线下面积AUC衡量区分能力。0.5-0.7代表模型的区分能力差，这个模型不太适用。0.7-0.9这个模型的区分能力较好，已经能够应用在医学领域。而大于0.9则表明这个模型的区分效果很好。而校正度(calibration)则是，你得出的预测概率是否和真实的结果一致性评价。我们一般用Hosmer-Lemeshow (H-L) test评价，所得统计量卡方值越小、对应P值越大校准度越好。最后的DCA就模型进行预测时是否真正的能够净获益。我们首先进行区分度的检验，而模型的区分能力，具有两个评判指标，分别是ROC曲线下面积和C指数(C-index)，对于logistic回归分析来说，我们使用ROC曲线下面积(AUC)这一个指标就可以了。

插上一句，在进行模型的评价时，我们如果用训练集training_set来评价模型显然并不合理，这未免就有王婆卖瓜自卖自夸的嫌疑了。所以，就需要我们的test_set了。首先我们使用训练集所得到的模型对测试集的每一个观测进行预测。得出每一个观测出现结局事件的预测概率，我们展现前面的十个预测值的结果。

y_pred = predict(classifier, newdata = test_set[-4],type = "response")

y_pred[1:10]

2 4 5 9 12 14

0.023999387 0.009636432 0.003276809 0.002999543 0.006156426 0.010334937

18 19 20 22

0.293941724 0.368247383 0.438425694 0.564420975

我们可以看到前面十个的预测值的结果其实并不是返回的0或者1，而是返回一个个小数，这是什么呢?其实这代表通过模型进行预测后，预测结局变量为阳性结果的可能性。接下来我们进行ROC曲线的计算，这里我们使用时ROC的pROC包。没有安装的的同学记得安装一下。

##install.packages(“ pROC ”)

library(pROC)

Type 'citation("pROC")' for a citation.

##

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

##

cov, smooth, var

```
ROC=roc(test_set[,4],y_pred)
```

Setting levels: control = 0, case = 1

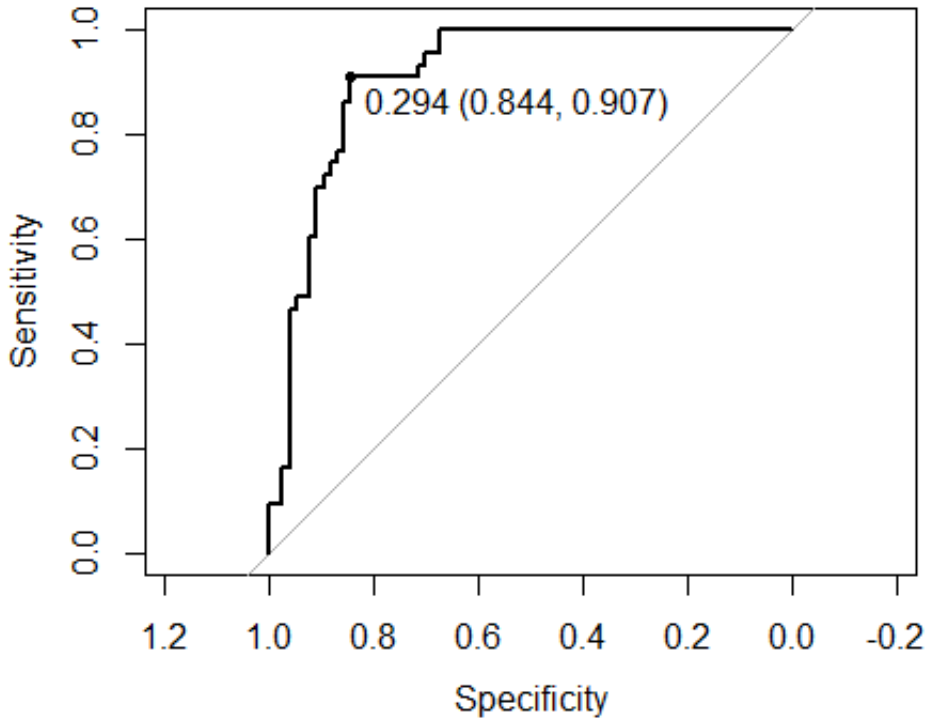
Setting direction: controls < cases

计算ROC曲线下面积和最佳阈值，并绘图。

```
auc(ROC)
```

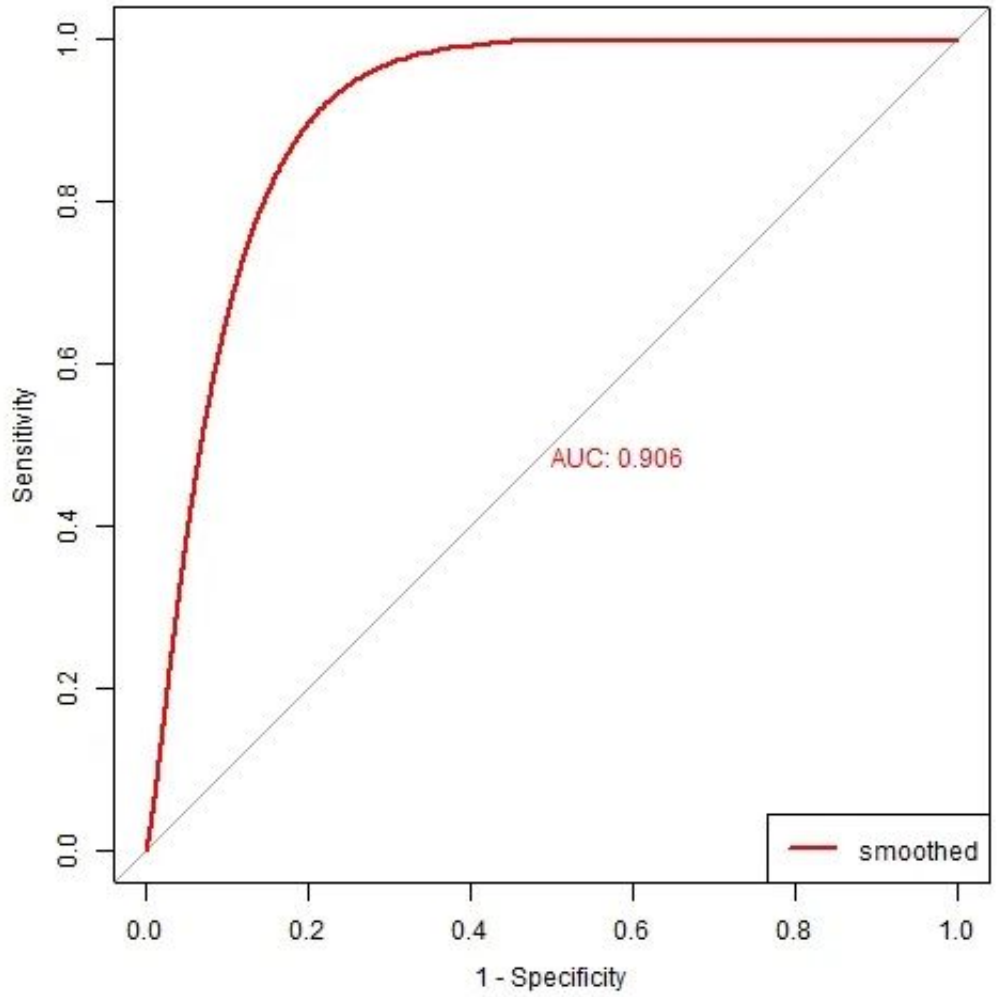
Area under the curve: 0.9103

```
plot(ROC,print.thres=T,print.thres.best.method="youden")
```



这样就绘制出了我们的ROC曲线，得到曲线下面积是0.9103，根据我们之前所说的这个模型很棒了对不对。并且我们打印出其最佳的阈值，也就是0.2940，其特异度和灵敏度分别是0.844，0.907。这里的最佳阈值即是约登指数最高的值，等于也就是灵敏度与特异度之和减去1。也就是曲线最左上角的那个点。接下来我们稍微美化一下图片

```
plot(smooth(ROC),col="red",print.auc=T,legacy.axes=T)

legend("bottomright",legend = c("smoothed"),col = "red",lwd = 2)
```



这样是不是好看多了呢?

（六）Calibration和DCA

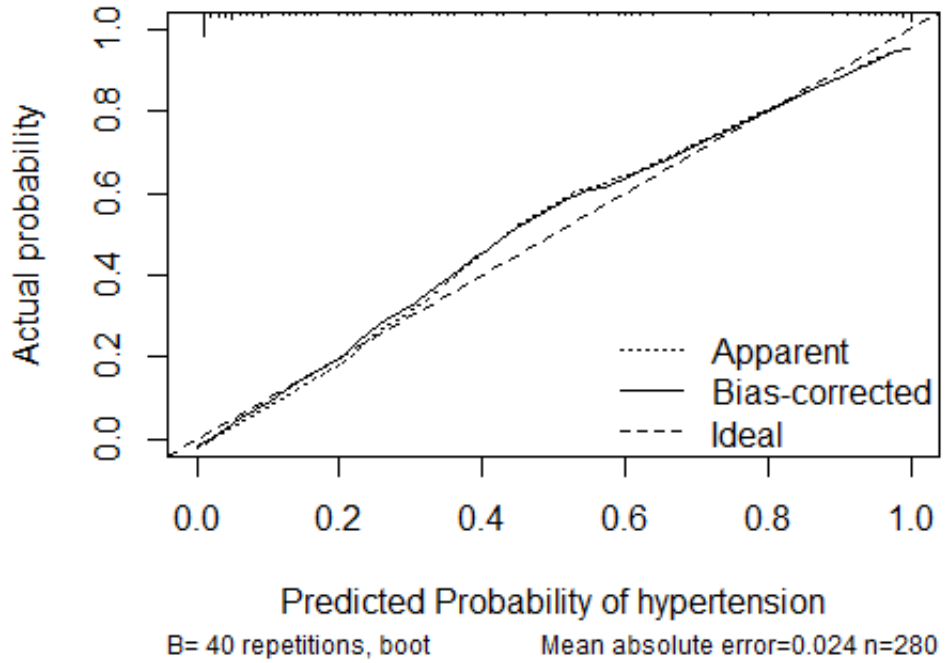
接下我们要讲Calibration和DCA了。大家注意听喔。首先重申一下对于模型的Calibration，其主要目的是评估我们预测得到的预测值与实际的真实性的一致性。

通俗的说就是我们的到的预测值和真实值存在多少偏差，这个偏差能不能接受?

这一步我们分为两个小步，第一步就是绘制我们熟悉的calibration图(我将给大家介绍两个方法)，第二步Hosmer-Lemeshow good of fit test(拟合优度检验)来进行相关的评估得到P值。

Ok我们先开始，加载我们需要的包rms包的功能非常强大,里面的两个函数功能都能达到矫正的作用，但是它的方法是不一样的。我们先上数据。

```
###data processing

dataset = read.csv('disease.csv')

dataset = dataset[2:5]

dataset$hypertension = factor(dataset$hypertension, levels = c(0, 1))

library(caTools)

set.seed(123)

split = sample.split(dataset$hypertension, SplitRatio = 0.7)

training_set = subset(dataset, split == TRUE)

test_set = subset(dataset, split == FALSE)

###construct the model

options(datadist=NULL)

classifier=lrm(hypertension~.,data=training_set,x=T,y=T)

###method 1

calib<-calibrate(classifier,method = "boot") ##calIbrate对模型进行评价 ###画图

plot(calib,lwd=2,lty=1,

xlim=c(0,1),ylim = c(0,1),

xlab = "Predicted Probability of hypertension",

ylab = "Actual probability",

col=c(rgb(192,98,83,maxColorValue = 255)))
```



##

n=280 Mean absolute error=0.024 Mean squared error=0.00088

0.9 Quantile of absolute error=0.053

###method 2

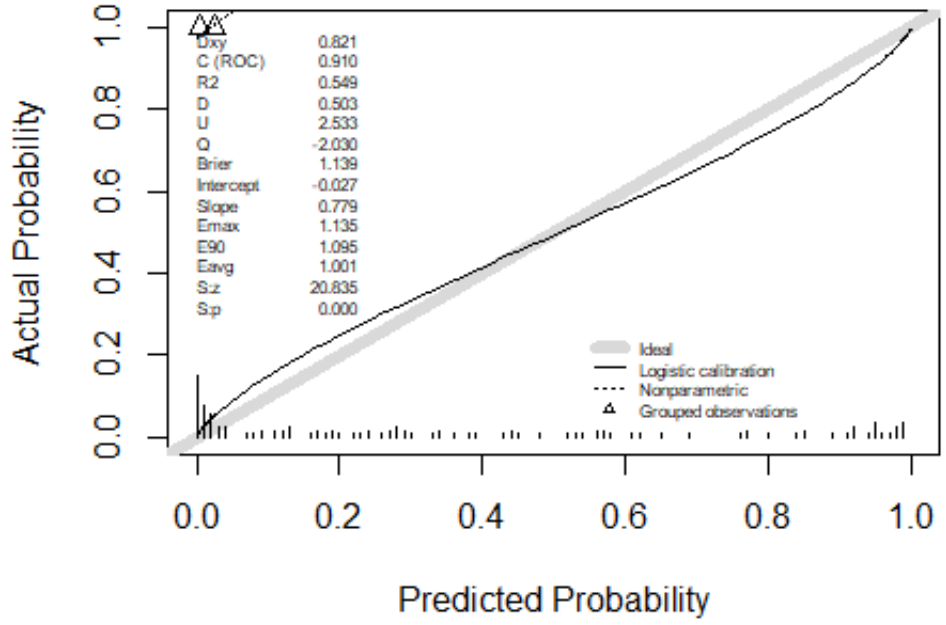
pred.logit = predict(classifier, newdata = test_set[-4])

##得到预测值，由于是logistics回归，故需要进行logit转换

y_pred <- 1/(1+exp(-pred.logit))

#####val.prob进行校正

val.prob(y_pred,as.numeric(test_set$hypertension),m=20,cex=0.5)



Dxy C (ROC) R2 D D:Chi-sq

8.205980e-01 9.102990e-01 5.492686e-01 5.030060e-01 6.136072e+01

D:p U U:Chi-sq U:p Q

NA 2.532954e+00 3.059545e+02 0.000000e+00 -2.029948e+00

Brier Intercept Slope Emax E90

1.138589e+00 -2.709633e-02 7.785694e-01 1.135061e+00 1.094938e+00

Eavg S:z S:p

1.000890e+00 2.083508e+01 2.081675e-96

第一步：绘制calibration图

好，我们使用两种方式都能得到calibration的图。提一句rms里面的这两种功能都需要使用rms的内置功能建模，也就是lrm进行建模才行。R本身自带的glm不能达到效果。好，接下来我们来看看两种方式的不同。

第一种

我们首先看calibration里面有一个boot，这个其实就是代表这个功能进行calibration的核心思想，也就是重抽样的思想。说起这个也是一个趣事，重抽样(bootstrap)本身的思想很简单，它和交叉验证(cross-validation)是机器学习非常重要的两大思想。重抽样可以理解为，你想知道池塘里面有多少条鱼，你先打捞上来100条，做好标记后放回去。然后然捞上来100条，根据这100条里有多少是你标记的鱼来推测总量，如此反复后不就知道这个池塘有多少条鱼了吗。

之前我说的趣事也就是当年这个思想的开发者写好论文向统计学的某某大牛期刊投稿时，期刊编辑给出了拒绝，并说“ 想法很好，但过于简单 ”。但是后面的实践发现bootstrap的作用远超大家的想象。现在想想，大牛发现这么有创造性的方法都能被拒，我们被拒几次岂不是稀松平常?哈哈哈哈，不吹牛我们继续。接下来我们看一看图的X轴和Y轴，X轴代表我们预测的可能性，而Y轴代表的是实际的可能性。理想状态我们的Calibration曲线应该是斜率为1从原点穿过的直线，也就是图中的虚线。我们可以看到我们的黑线，也就是实际的线和虚线差不太多，证明我们的结果还是可以的哈。并且平均的总的误插在0.024，还不错!

第二种

val.prob，其实就是我们传统认知的calibration的方法，根据我们预测出来的预测概率与真实值进行矫正。我们看到这个val.prob和calibate功能不同的是它出现了很多其他奇奇怪怪的指标。我们主要关注四个指标Dxy, C(ROC), U: p, Brier.Dxy即是我们的预测值和真实值之间的相关性，C(ROC)即是ROC曲线下面积。U检验本身是unrealiability检验，即是假设预测值和真实值两者之间没有相关性。我们发现U:p很小，接近于0。那么则提示我们有很好的校正度。最后一个brier即是预测值和真实值的平均平方差，这个值当然越小越好。如果大家还想知道更多。像我一样help一下

help("calibrate")

starting httpd help server ... done

help("val.prob")

第二步：Hosmer-Lemeshow检验得到校准P值

Hosmer-Lemeshow检验得到校准P值，其实HL检验是目前统计上用的比较多的检验校准度的方法。其思想是：

1.首先根据预测模型来计算每个个体未来发生结局事件的预测概率;

2.根据预测概率从小到大进行排序，并按照十分位等分成10组;

3. 分别计算各组的实际观测数和模型预测数，其中模型预测数，即每个人的预测概率*人数，再求总和，这里人数即为1，最后总和就相当于每个个体预测概率的直接加和;

4. 根据每组实际观测数和模型预测数计算卡方值(自由度=8)，再根据卡方分布得到对应的P值。

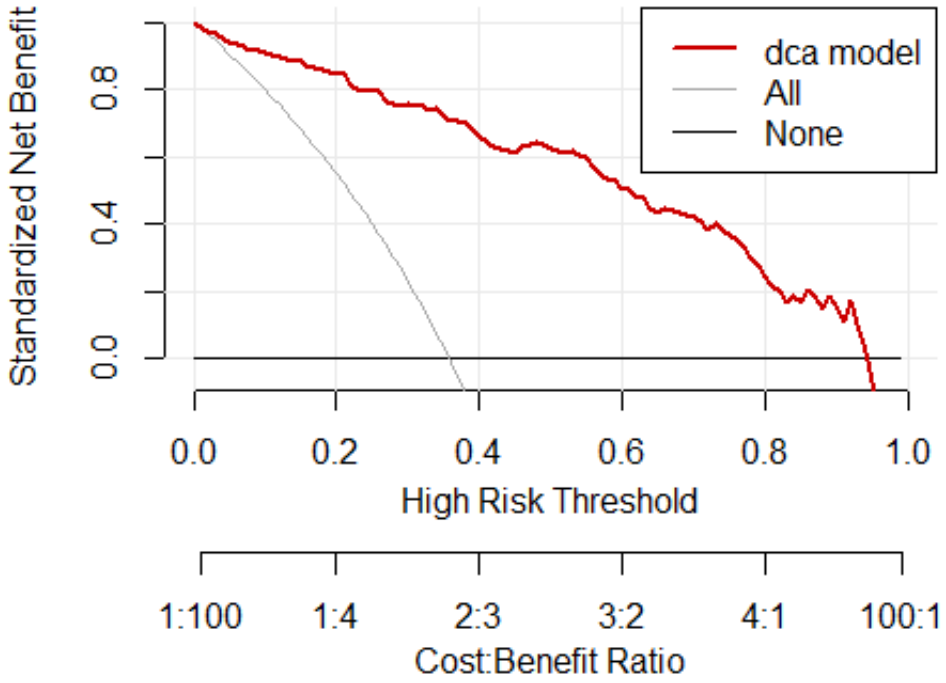
大家看懂了吗?没事就算没看懂也不耽误我们来用是吧。我们直接看结果。如果所得的统计量卡方值越小，对应的P值越大，则提示预测模型的校准度越好。若检验结果显示有统计学显著性(P<0.05)，则表明模型预测值和实际观测值之间存在一定的差异，模型校准度差。好理论有了，上

代码。

```
###install.packages("RsourceSelection")

library(ResourceSelection)

## ResourceSelection 0.3-5 2019-07-22

hoslem.test(as.numeric(test_set$hypertension),y_pred,g=10)

##

## Hosmer and Lemeshow goodness of fit (GOF) test

##

## data: as.numeric(test_set$hypertension), y_pred

## X-squared = 6533.2, df = 8, p-value < 2.2e-16
```

好了，我们的Calibration弄完了，接下来我们要到最后的部分，也是最加分的部分，DCA。在上一篇我大概讲了什么是DCA，因为理论太多，我们先学会用，理论我们慢慢的学都可以。首先我们还是得加载包，rmda包是我们依赖得做出DCA得包，所以我们必须得加载它。

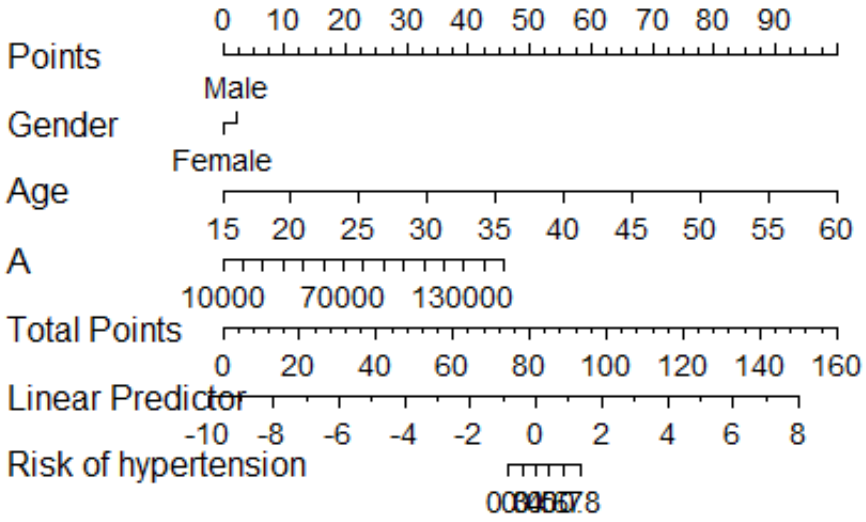
```
#install.packages(“ rmda ”)

library(rmda)

training_set[,4]=as.numeric(training_set[,4])-1

dca=decision_curve(formula = hypertension ~Gender+Age+A,

data = training_set,family = "binomial",

confidence.intervals = F,bootstraps = 10,fitted.risk = F)

## Note: The data provided is used to both fit a prediction model and to estimate the respective decision curve. This may cause bias in decision curve estimates leading to over-confidence in model performance.

plot_decision_curve(dca,curve.names = "dca model")
```



红色得曲线即是我们DCA构建模型得到曲线，如果我们的红线在水平的黑线以及左侧斜向的灰线以上，那么我们可以认为这一段的红线取值是能够获益的。而我们的模型，红线从几乎0到1都在灰线和黑线之上，我们可以认为根据我们所构建模型所作出得决策能够给病人带来获益，因此这是一个相当完美得模型。

大家祝贺一下!希望我们自己文章的模型也能有这么完美!!!那么整体的内容我们已经弄完了，不过大家觉得是不是差点意思?大家觉得还可以，不差吗?但我觉得可能还差点意思。来，我们补上。

```
ddist <- datadist(training_set)

options(datadist='ddist')

nomogram1 <- nomogram(classifier,

fun=function(x)1/(1+exp(-x)), # or fun=plogis

fun.at=seq(0.3,0.8,by=0.1),

funlabel="Risk of hypertension")

plot(nomogram1)
```



这个图大家是不是很熟悉，对，就是我们得nomogram。我们每一个变量根据它的大小我们都能得到一个评分，并且根据每个变量进行相加。

OK,我们的logistic模型的评价就做完了。谢谢大家！