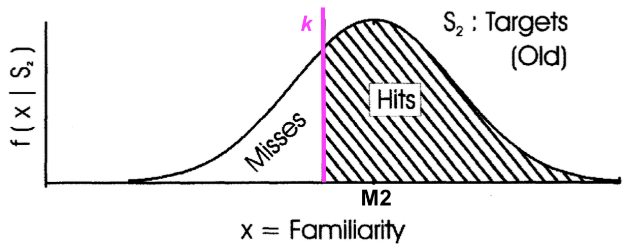
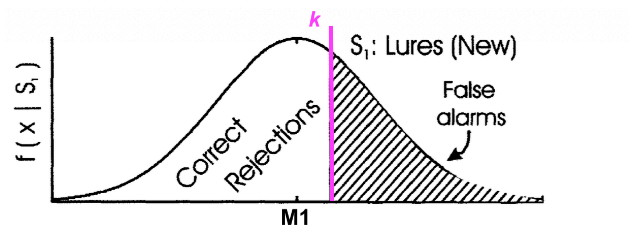
L'esperimento "Yes-No": Il partecipante come classificatore binario

Nella precedente lezione abbiamo imparato a misurare la sensibilità del partecipante nei compiti di discriminazione (ad es., *stimolo vecchio/nuovo, stimolo A/B, segnale/rumore ecc. ecc).* Il fondamento teorico delle misure di accuratezza invariati rispetto al bias di risposta (***d'* e curva ROC**) è la Teoria della Detezione del Segnale (in inglese, SDT).

Chiediamoci ora quale sia il modello psicologico del processo decisionale che ha luogo, ad esempio, nel compito di riconoscimento di volti vecchi/nuovi. La SDT assume che il partecipante ad un simile esperimento di memoria ricerchi nello stimolo (un volto vecchio o nuovo) un "segnale*"*, che potremo chiamare in questo specifico contesto "familiarità" per un volto.



Ciascun valore di *segnale-familiarità* ha un certa probabilità di originarsi dagli stimoli costituiti dai volti vecchi, già visti precedentemente dal partecipante. In media, la familiarità per i volti vecchi sarà superiore a quella per i volti nuovi, altrimenti il partecipante non sarebbe in grado di discriminare accuratamente: ciò determina uno spostamento verso destra della distribuzione rispetto alla distribuzione dei valori di familiarità per i volti nuovi (**M2 > M1**). Inoltre, alcuni volti nuovi genereranno valori di familiarità sovrapponibili a quelli dei volti vecchi, risultando pertanto ambigui e rendendo la prestazione non perfetta.



Questo modello a due distribuzioni (normali) rappresenta il problema decisionale "interno" del partecipante, il quale può valutare il *segnale-familiarità* dello stimolo senza sapere quale distribuzione lo abbia generato. La regola decisionale ottimale (*non perfetta*) secondo la SDT è quella di stabilire un *criterio* (k, riportato nella figura in violetto) che divida l'asse della familiarità in due parti: (1) sopra il criterio il partecipante risponderà "YES" (*la faccia è sufficientemente familiare da essere vecchia*), (2) al di sotto risponderà invece "No" (*la faccia non è sufficientemente familiare

e potrebbe essere nuova). Il partecipante si comporterà quindi come un **classificatore binario**, decidendo se classificare una faccia come vecchia ("YES") o nuova ("No"), usando il modello decisionale basato sul criterio k che verrà confrontato con l'intensità del segnale-familiarità.
In questo processo decisionale, gli Hit (H) rappresentano le proporzioni di risposte corrette "YES" ad uno stimolo vecchio, mentre le proporzioni di Falsi Allarmi (F) si originano da stimoli nuovi i cui livelli di familiarità hanno superato il criterio k, traendo in inganno il partecipante.

Si noti che:

I. il criterio k può essere volontariamente "spostato" dal partecipante verso destra (< bias "Yes": H e F —> 0.00) o verso sinistra (> bias "Yes": H e F —> 1.00),
II. questo spostamento volontario non altera le distanze tra la familiarità media percepita delle due classi di volti vecchi (M2) e nuovi (M1),
III. la distanza M2-M1 rappresenta la misura di *sensibilità* (d') alla discriminazione dei volti che secondo la SDT (Vedi Lezione 18, pp. 18-20).

L'indice AUC - "Area Under (the ROC) Curve"

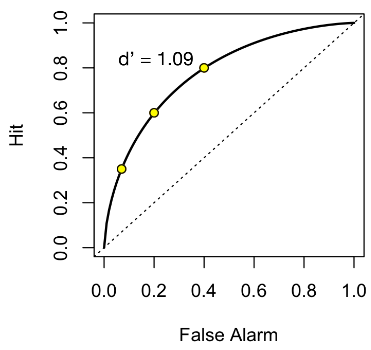
La prestazione del partecipante, riassunta nella tabella stimolo-risposta

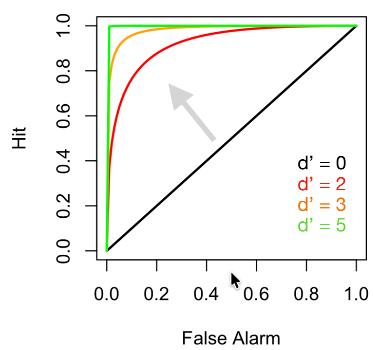
Stimulus Class	*Response*		
	"Yes"	"No"	Total
Old (S_2)	.8	.2	1.0
New (S_1)	.4	.6	1.0

può essere quantificata in termini di sensibilità (invariante rispetto al criterio k) con l'indice:

$$d' = z(H) - z(F),$$

il cui locus geometrico è la curva di isosensitività (o ROC) costituita dalle coppie (F,H) che generano lo stesso valore d'. Ad esempio, le coppie (.4,.8), (.2,.6) e (.07,.35) determinano un d'=1.09 (punti gialli in figura), differiscono tra loro solo per il criterio k progressivamente più restrittivo (< bias "Yes") e si trovano tutte sulla medesima curva ROC. Quando la riclassificazione binaria "stimolo vecchio: Si-No" è casuale, d'=0, la ROC è la diagonale maggiore (*chance line*), dove le proporzioni di H e F coincidono. Al crescere della sensibilità al segnale di familiarità dei volti, la curva ROC si sposta verso l'angolo in alto a sinistra dove l'accuratezza è perfetta (H = 1, F = 0).





Dal momento che gli assi dello spazio ROC vanno da 0 a 1, la regione in cui possono estendersi le curve di isosensitività è un quadrato di area unitaria: ciò significa che a un d'=0 corrisponderà metà dell'area dello spazio ROC, diremo quindi che la sua "area sotto la curva ROC" avrà valore AUC = 0.5. Valori crescenti di d' inscriveranno sotto la curva ROC quantità maggiori di AUC, fino al limite superiore dove per d'>4 avremo valori AUC approssimativamente uguali a 1. La conversione tra d' e AUC si basa sull'assunzione che le distribuzioni di probabilità del segnale e del rumore siano gaussiane con varianza omogenea (…e media diversa naturalmente):

$$AUC = \Phi\left(\frac{d'}{\sqrt{2}}\right),$$

dove Φ è l'area sotto l'ogiva gaussiana standard al punto $d'/\sqrt{2}$, dalla quale si ricava agevolmente la formula inversa $d' = \sqrt{2} \cdot \Phi^{-1}(AUC)$, dove Φ^{-1} è il punto z normale standard corrispondente all'area cumulata AUC.

$$\begin{aligned}
d' &= 0 && \rightarrow AUC = \Phi(0) = .50 \\
d' &= 1.09 && \rightarrow AUC = \Phi(0.771) = .80 \\
d' &= 2 && \rightarrow AUC = \Phi(1.414) = .92 \\
d' &= 4 && \rightarrow AUC = \Phi(2.83) \approx 1
\end{aligned}$$

L'area sotto la curva ROC fornisce quindi una misura della capacità di discriminazione di un classificatore binario (…soggetto umano o modello statistico che sia), definibile anche come "potenza predittiva" essendo vincolata tra 0 e 1. Come regola generale possiamo definire i seguenti valori di soglia:

AUC = 0.50	Prestazione casuale, nessuna capacità di discriminazione.
0.70 ≤ AUC < 0.80	Accettabile (…moderata)
0.80 ≤ AUC < 0.90	Eccellente (…alta)
0.90 ≤ AUC < 1	Eccezionale, discriminazione *quasi* perfetta.

Il modello Logit come classificatore binario.

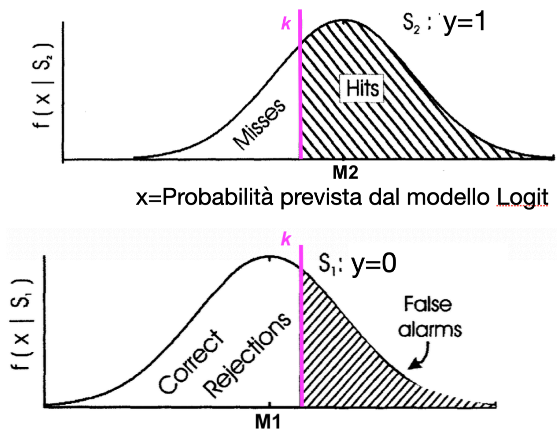
✦ Il modello di regressione logistica (Logit) offre la soluzione predittiva di massima verosimiglianza per una variabile dipendente binaria $y_i = (0,1)$ dove ad esempio 0=*"insuccesso"/"assenza"/"no"* e 1=*"successo"/"presenza"/"si"*.

✦ Il modello raggiunge una *sensibilità* alla presenza del segnale $y_i = 1$ quando è efficacemente "sintonizzato" sul problema oggetto dello studio, grazie ad una serie di covariate x_{ij} statisticamente significative.

✦ In analogia con lo spazio decisionale del riconoscimento di volti vecchi e nuovi, dove il segnale percepito da un volto veniva trasformato psicologicamente in una quantità continua, chiamata "familiarità del volto", possiamo raffigurare anche per il modello Logit uno spazio decisionale simile, anche se naturalmente di natura "matematica".

✦ In questo contesto, il segnale presente nello stimolo (*una persona con pensiero suicidario, con un "vissuto" personale di poliabuso*) viene colto grazie al valore delle covariate x_j e trasformato, mediante il modello di regressione logistica, in una quantità continua di "probabilità y=1".

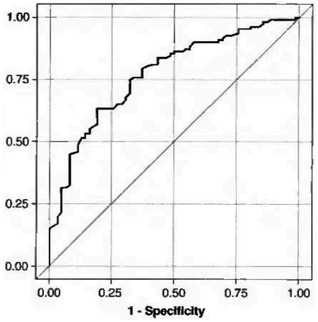
✦ Continuando con l'analogia, la strategia decisione (*dell'operatore che utilizzerà il modello*) sarà quella di stabilire un criterio k di "probabilità soglia" oltre il quale affermare che, dati certi valori di input x_j, il valore previsto per y sarà 1 (ad es., *pensiero suicidario, rischio di tentato suicidio,…*) oppure 0 (*nessun rischio*).



✦ Essendo il modello Logit (come ogni altro modello statistico) non perfettamente adattato ai dati, (ricordiamoci infatti che solo un modello saturo lo sarebbe), sarà soggetto ad un errore di previsione che renderà la sua accuratezza discriminativa (sensibilità) non assoluta e costante rispetto ai valori di input x_{ij}.

✦ Analogamente, l'osservatore umano che classifica i volti decidendo se siano volti vecchi o nuovi affidandosi alla sua sensibilità (ovvero la distanza stimata M2-M1 tra le distribuzioni di familiarità per volti vecchi e nuovi) e ad un criterio k, potrebbe non avere una percezione costante della qualità di familiarità lungo l'esperimento. Se ciò avviene, al modificarsi del criterio k (ad es., per ragioni sperimentalmente indotte) si osserva come la misura d' non sia costante ma piuttosto "stabile", attorno ad un valore medio, che genera coppie (F,H) solo approssimativamente curvilinee.

✦ Il grafico risultante sarà chiamato **ROC empirica**.



- Ritornando al modello Logit, essendo un modello statistico che cerca di utilizzare al meglio l'informazione contenuta nelle covariate disponibili, per generare un segnale imperfetto di "probabilità y=1", anch'esso genererà un valore *d'* solo "stabile" e non costante per tutte le coppie (F,H) originate da uno spostamento del criterio k di "probabilità soglia".

- Le curve ROC empiriche generate dai modelli logit sono spesso funzioni a gradini poiché sono costruite a partire da campioni di dati "finiti", anziché da distribuzioni teoriche continue. Sebbene un modello logit preveda probabilità continue, la curva ROC empirica viene costruita spostando una soglia decisionale k, ma i tassi di H e F cambiano solo a soglie specifiche, corrispondenti alle proporzioni empiriche previste dal campione. In pratica, essendo il modello Logit stimato su un campione finito di dati "effettivamente positivi" ed "effettivamente negativi", spesso limitati o basati su eventi y rari, le proporzioni H e F cambiano per salti discreti (gradini) anziché in modo graduale.

- Inoltre, quando il modello produce probabilità previste identiche per più osservazioni, la modifica della soglia k di classificazione determina un travaso più ampio di veri/falsi positivi contemporaneamente, creando gradini verticali o diagonali nel grafico. A differenza della curva ROC teorica, che presuppone una distribuzione sottostante continua, la curva ROC empirica è non parametrica e riflette la natura discreta/finita dei dati.

Vediamo con R quanto discusso, utilizzando i dati sul suicidio vs. abuso di sostanze.

✤ Ricerchiamo la directory della sessione corrente di R e mettiamo qui il file RData:
```
getwd()
```

✤ Carichiamo il data set:
```
load("dataset08.RData")
```

✤ Regressione logistica: pensiero suicidario
```
Mod.PS_08<-glm(MHSUITHK ~ IRSEX + NDSSDNSP + DEPNDALC + DEPNDCOC + DEPNDHER + DEPNDANL +
 DEPNDSED + DEPNDSTM + DEPNDHAL + DEPNDINH + DEPNDMRJ + DEPNDTRN, data =
 dataset08_reduced, family=binomial)
```

✤ Regressione logistica: pianificazione suicidaria
```
Mod.PianS_08<-glm(MHSUIPLN ~ IRSEX + NDSSDNSP + DEPNDALC + DEPNDCOC + DEPNDHER + DEPNDANL
 + DEPNDSED + DEPNDSTM +DEPNDHAL + DEPNDINH + DEPNDMRJ + DEPNDTRN, data =
 dataset08_reduced, family=binomial)
```


✤ Regressione logistica: tentato suicidio
```
Mod.Tent_08<-glm(MHSUITRY ~ IRSEX + NDSSDNSP + DEPNDALC + DEPNDCOC + DEPNDHER + DEPNDANL
 + DEPNDSED + DEPNDSTM +DEPNDHAL + DEPNDINH + DEPNDMRJ + DEPNDTRN, data =
 dataset08_reduced, family=binomial)
```

✤ Calcoliamo le probabilità previste dal modello per il pensiero suicidario
```
prob = predict(Mod.PS_08, dataset08_reduced,type="response")
prob[1:6]
# 0.05115771 0.03653189 0.03653189 0.09952246 0.05115771 0.05115771
length(prob)
# [1] 37357
```

✤ Impostiamo un criterio decisionale k per la probabilità soglia "y=1" e assegniamo a ciascuna osservazione valore "1" se prob > k
```
k = 0.10085327
Classificazione.binaria<-ifelse(prob>k,1,0)
```

✤ Costruiamo la tabella stimolo-risposta
```
matching.st.risp <-rep(NA,length(prob))
for(i in 1:length(matching.st.risp)){
if(dataset08_reduced$MHSUITHK[i]==0 & Classificazione.binaria[i]==0)
 matching.st.risp[i]<-"CorrRej"
if(dataset08_reduced$MHSUITHK[i]==0 & Classificazione.binaria[i]==1)
 matching.st.risp[i]<-"F"
if(dataset08_reduced$MHSUITHK[i]==1 & Classificazione.binaria[i]==1)
 matching.st.risp[i]<-"H"
if(dataset08_reduced$MHSUITHK[i]==1 & Classificazione.binaria[i]==0)
 matching.st.risp[i]<-"Miss"
                                                }
Tab.st.risp<-table(matching.st.risp)
Tab.st.risp
# matching.st.risp
# CorrRej       F       H    Miss
#   33709    1562     393    1693
H<-Tab.st.risp[3]/(Tab.st.risp[3]+ Tab.st.risp[4])
F<-Tab.st.risp[2]/(Tab.st.risp[2]+ Tab.st.risp[1])
dprime=qnorm(H)-qnorm(F);
names(dprime)="d'"
```

✤ Visualizziamo i risultati
```
dprime;F;H
##         d'
## 0.8191705
##          F
## 0.04428567
##         H
## 0.1883988
```

✤ La procedura può essere automatizzata in R mediante il pacchetto "pROC"
```
install.packages("pROC",dep=TRUE)
library(pROC)

## Stima automatizzata delle coppie (F,H) per diversi criteri k:
ROC_<-roc(dataset08_reduced$MHSUITHK,prob,smooth=FALSE,ci=TRUE)
```

✤ Il comando roc() genera diversi valori di probabilità-soglia k, detti "thresholds", le relative proporzioni empiriche di HIT = "sensitivity" e Correct Rejections = "specificity", dalle quali ricaviamo i False-Alarm = 1- "specificity":

```
## valori probablità soglia "k"
ROC_$thresholds
## Hit (o "sensitivity"):
ROC_$sensitivities
## Falsi allarmi (o 1-specificity):
1-ROC_$specificities
```

✤ La soglia numero 19 coincide con il nostro precedente k = 0.10085327.

```
ROC_$thresholds[19]
[1] 0.1008533
```

✤ Visualizziamo la coppia (F,H) relativa a questo valore soglia, che sarà naturalmente uguale alla coppia precedentemente identificata (0.04428567, 0.1883988):

```
ROC_$sensitivities[19]
## [1] 0.1883988
1-ROC_$specificities[19]
## [1] 0.04428567
dprime
       d'
0.8191705
qnorm(ROC_$sensitivities[19])-qnorm(1-ROC_$specificities[19])
## [1] 0.8191705
```

✤ Scegliendo diversi valori per k si generano diverse coppie (F,H), il valore d' associato a queste coppie non è costante, come ci si aspetterebbe dalla teoria SDT, in quanto le coppie (F,H) si dispongono su ROC empiriche solo approssimativamente curvilinee.

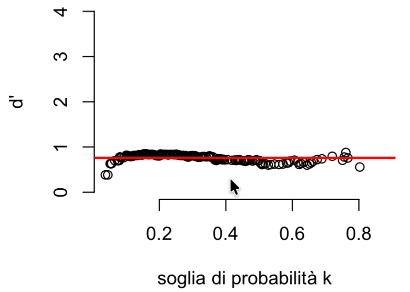
```
## Scegliamo ad esempio il criterio k = 0.17938
ROC_$thresholds[62]
## [1] 0.17938
ROC_$sensitivities[62]
## [1] 0.08197507
1-ROC_$specificities[62]
## [1] 0.01309858
## d-prime leggermente diverso...
qnorm(ROC_$sensitivities[62])-qnorm(1-ROC_$specificities[62])
[1] 0.8313681

## Scegliamo ad esempio il criterio k = 0.38964615
ROC_$thresholds[136]
## [1] 0.3896462
ROC_$sensitivities[136]
## [1] 0.01246405
1-ROC_$specificities[136]
## [1] 0.001502651
## d-prime ancora leggermente diverso...
qnorm(ROC_$sensitivities[136])-qnorm(1-ROC_$specificities[136])
[1] 0.7246799

## Tutti d' diversi!
plot(x=ROC_$thresholds, y=qnorm(ROC_$sensitivities)-qnorm(1-ROC_$specificities),
```

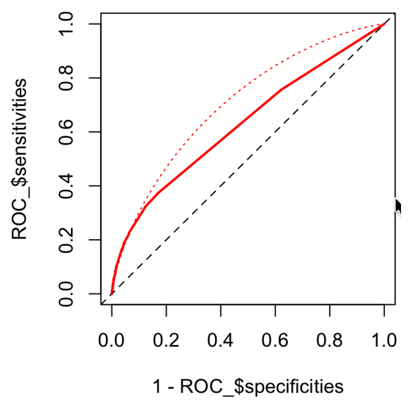


```
     ylim=c(0,4),xlab="soglia di probabilità k",ylab="d'",bty="n")
dprime<-qnorm(ROC_$sensitivities)-qnorm(1-ROC_$specificities)
summary(dprime)
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NAs
##   -Inf  0.7086  0.7930    -Inf  0.8248  0.8793       2
mean_dprime<-mean(dprime[!is.na(dprime) & !is.infinite(dprime)])
mean_dprime
## [1] 0.7619664
abline(h=mean_dprime,col="red",lwd=2)
```



✤ Ne consegue, come già anticipato, che il locus delle coppie (H,F) generate dal modello Logit e basate su diversi valori k sarà solo approssimativamente una curva ROC, che chiameremo curva ROC empirica:

```
par(pty = "s") # area di disegno "quadrata"
plot(x=1-ROC_$specificities,y=ROC_$sensitivities,col="red",type="l",lwd=2)
abline(a=0,b=1,lty="dashed")
## sovrapponiamo la cruva ROC teorica per d' = 0.7619664
F=seq(0,1,by=0.01)
H=pnorm( 0.7619664 + qnorm(F) )
points(x=F,y=H,col="red",type="l",lty="dotted")
```



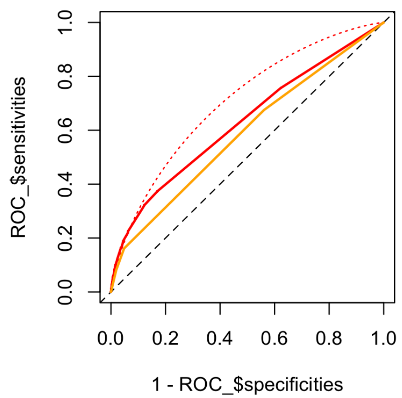
CURVE ROC - Applicazione nel modello Logit.

✤ Non essendoci un valore d' costante, ma soltanto "medio", valuteremo la capacità discriminativa del modello Logit utilizzando lo spazio ROC nella sua interezza, mediante l'area sotto la curva (AUC):

```
ROC_<-roc(dataset08_reduced$MHSUITHK,prob,smooth=FALSE,ci=TRUE)
ROC_
## Area under the curve: 0.6288
## 95% CI: 0.6161-0.6415 (DeLong)
```

✤ Sovrapponiamo la curva ROC per un modello ridotto, contenente solo i predittori IRSEX + DEPNDALC + DEPNDINH:

```
Mod.PS_08_RID<-glm(MHSUITHK ~ IRSEX + DEPNDALC, data = dataset08_reduced,family=binomial)
prob_RID = predict(Mod.PS_08_RID, dataset08_reduced,type="response")
ROC_RID<-roc(dataset08_reduced$MHSUITHK,prob_RID,smooth=FALSE,ci=TRUE)
ROC_RID
## Area under the curve: 0.5864
## 95% CI: 0.5744-0.5984 (DeLong)
points(x=1-ROC_RID$specificities,y=ROC_RID$sensitivities,col="orange",type="l",lwd=2)
```



✤ Eseguiamo un test statistico per confrontare la aree AUC del modello completo e ridotto: essendo modelli "nested", basati sugli stessi dati, sceglieremo l'opzione paired=TRUE.

```
roc.test(ROC_,ROC_RID,paired=TRUE)

##          DeLong's test for two correlated ROC curves
##
##  data:  ROC_ and ROC_RID
##  Z = 8.9678, p-value < 2.2e-16
##  alternative hypothesis: true difference in AUC is not equal to 0
##  95 percent confidence interval:
##   0.03313952 0.05167649
##  sample estimates:
##  AUC of roc1 AUC of roc2
##   0.6287965   0.5863885
```