Sensitivity, specificity and ROC

PhD Flavio Lichtenstein

2071, July 22th

Two distributions - t-student test (t.test)

Here we will present 2 distributions and make them close togheter or far appart. The statistics that measures the distance between two normal distribuitons is t-student.

```
source("support.R")

d1 = rnorm(200, mean=1, sd=1)
d2 = rnorm(200, mean=3, sd=.5)

data = data.frame(d1, d2)
options(digits=5)
head(data)
```

```
##        d1     d2
## 1 0.89339 2.5116
## 2 1.57862 2.7429
## 3 1.59501 3.1589
## 4 1.88974 3.3520
## 5 2.19981 3.9098
## 6 1.09019 3.4837
```

```
tt <- t.test(d1, d2)
print(tt)
```

```
##
##  Welch Two Sample t-test
##
## data:  d1 and d2
## t = -26.3, df = 290, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.1970 -1.8909
## sample estimates:
## mean of x mean of y
##    1.0067    3.0506
```

```
print(paste("pvalue", tt$p.value))
```

```
## [1] "pvalue 9.13347497841806e-79"
```

```
print(paste("CI = [", paste(round(tt$conf.int,3), collapse = " to "), "] - confidence interval", sep=""))
```

```
## [1] "CI = [-2.197 to -1.891] - confidence interval"
```

```
diff = d1-d2
mu.diff = mean(diff)
ssd.diff = sd(diff)
sem.diff = ssd.diff / sqrt(length(diff))

print(paste("mean(difference) =", round(mu.diff,3) ))
```

```
## [1] "mean(difference) = -2.044"
```

```
print(paste("ssd(difference)  =", round(ssd.diff,3) ))
```

```
## [1] "ssd(difference)  = 1.077"
```

```
print(paste("SEM(difference)  =", round(sem.diff,3) ))
```

```
## [1] "SEM(difference)  = 0.076"
```

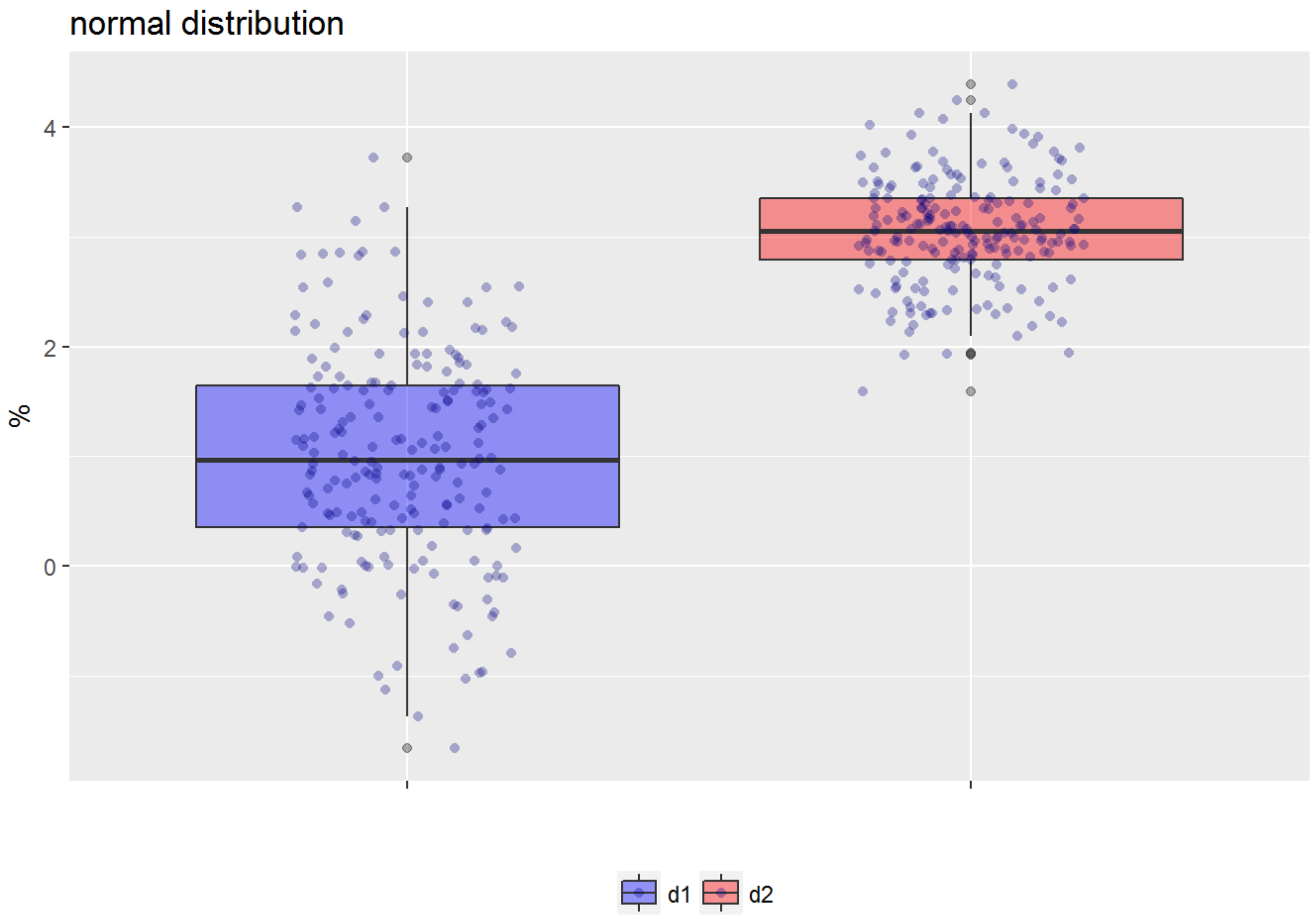
```
q.025 = qnorm(.025)

ci = mu.diff + c(1, -1) * q.025 * sem.diff
print(paste("CI calc = [", paste(round(ci,3), collapse = " to "), "] - confidence interval", sep=""))
```

```
## [1] "CI calc = [-2.193 to -1.895] - confidence interval"
```

```
print(paste("CI pack = [", paste(round(tt$conf.int,3), collapse = " to "), "] - confidence interval", sep=""))
```

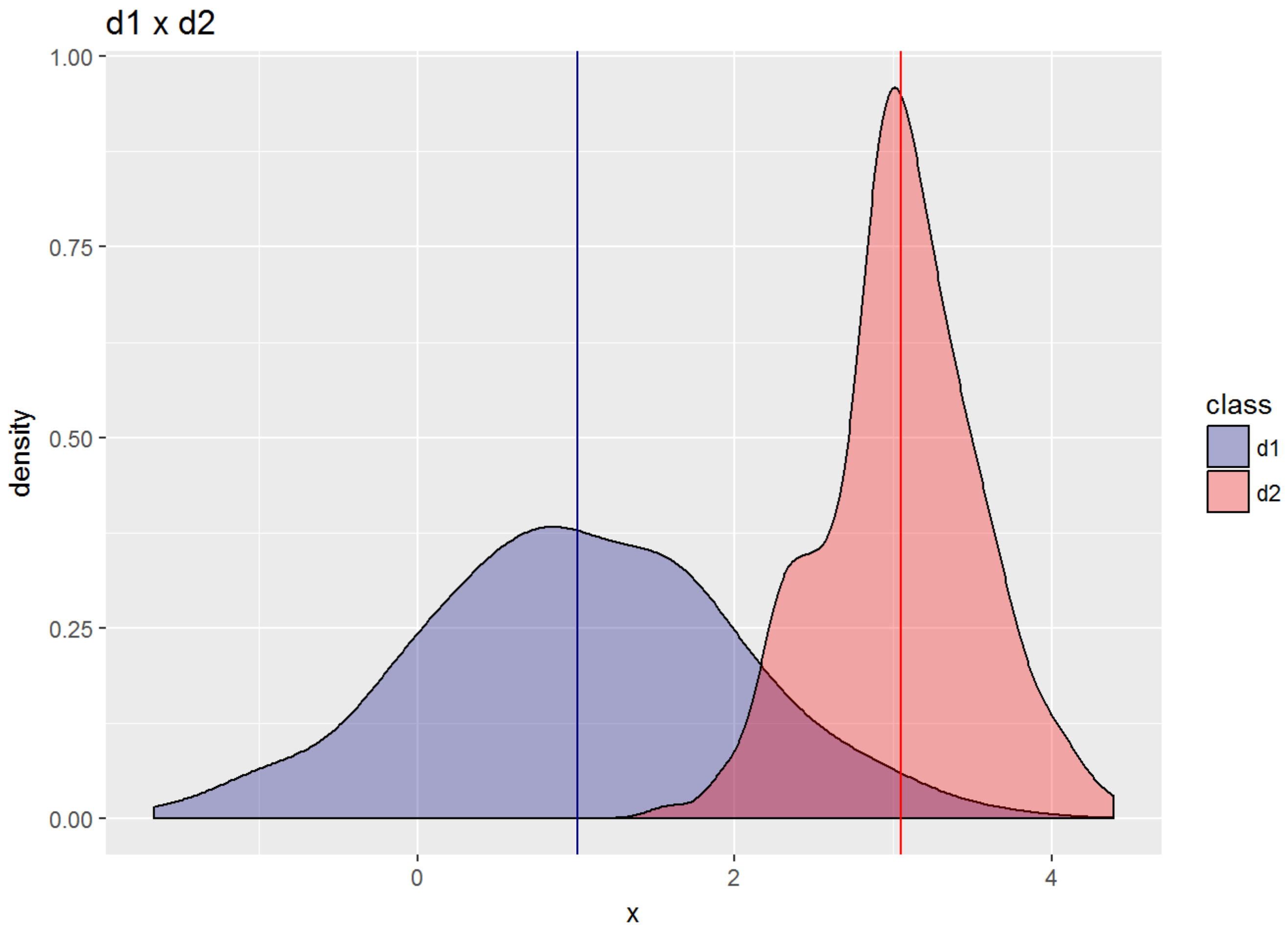
```
## [1] "CI pack = [-2.197 to -1.891] - confidence interval"
```

```
my.boxplot(data, classX=c("one","two"), cols=c(1,2), title="normal distribution", ylab="%", colors=c("blue", "red"),
is.log=F,ylim=NA)
```



Can we define if the distributions are sufficient far appart?

```
print_2densities(d1, d2)
```



```
healthy = rnorm(100, mean=1, sd=.7)
sick    = rnorm(100, mean=3, sd=.5)

healthy = round(healthy*100)
sick = round(sick*100)

tt <- t.test(healthy, sick)
print(tt)
```

```
##
##  Welch Two Sample t-test
##
## data:  healthy and sick
## t = -21.7, df = 176, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -213.97 -178.25
## sample estimates:
## mean of x mean of y
##    102.20    298.31
```

```
print(paste("pvalue", tt$p.value))
```

```
## [1] "pvalue 1.69577952372443e-51"
```

```
print(paste("CI = [", paste(round(tt$conf.int,3), collapse = " to "), "] - confidence interval", sep=""))
```

```
## [1] "CI = [-213.968 to -178.252] - confidence interval"
```

```
diff = healthy-sick
mu.diff = mean(diff)
ssd.diff = sd(diff)
sem.diff = ssd.diff / sqrt(length(diff))

print(paste("mean(difference) =", round(mu.diff,3) ))
```

```
## [1] "mean(difference) = -196.11"
```

```
print(paste("ssd(difference)  =", round(ssd.diff,3) ))
```

```
## [1] "ssd(difference)  = 86.276"
```

```
print(paste("SEM(difference)  =", round(sem.diff,3) ))
```

```
## [1] "SEM(difference)  = 8.628"
```

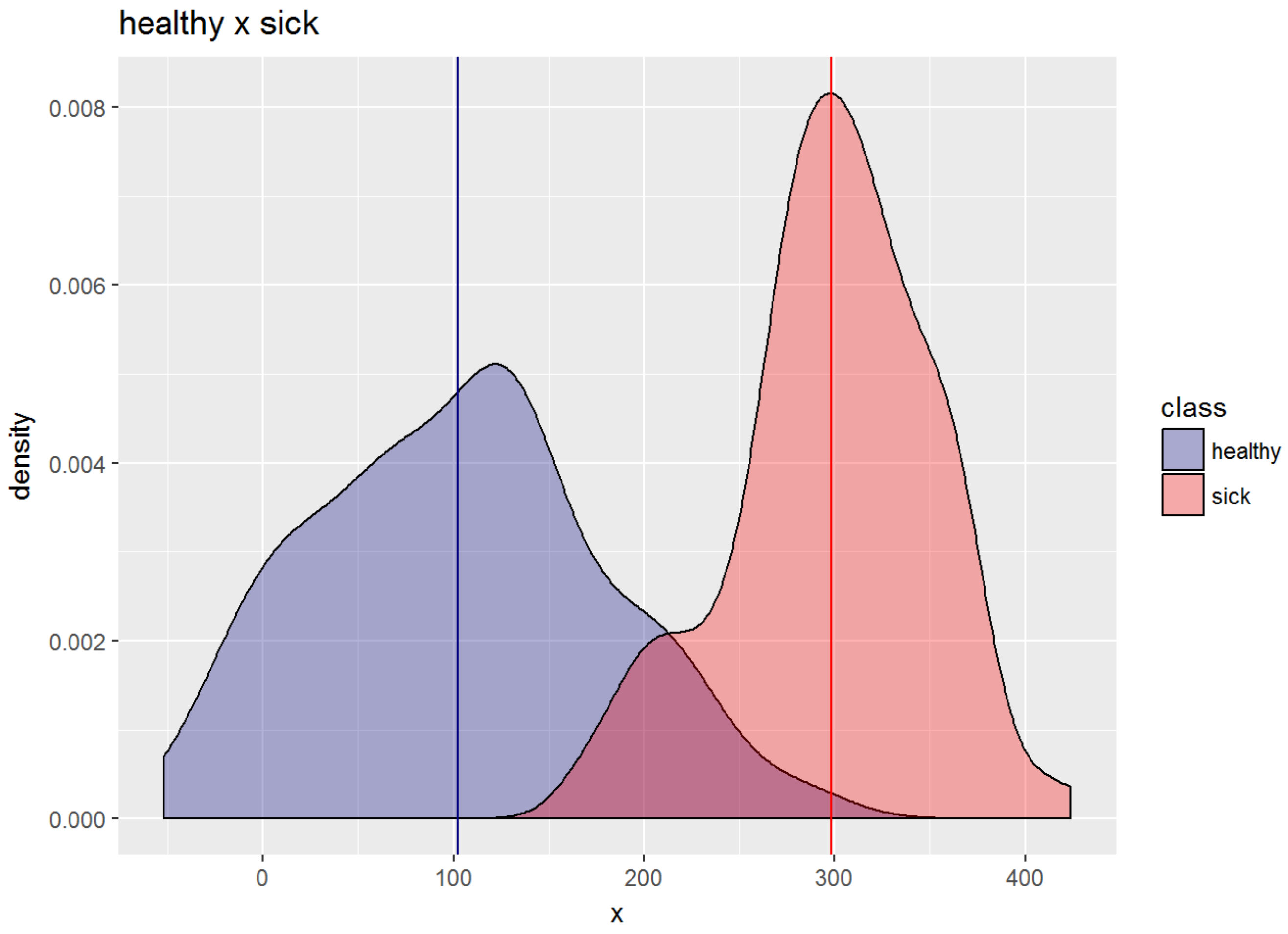
```
q.025 = qnorm(.025)

ci = mu.diff + c(1, -1) * q.025 * sem.diff
print(paste("CI calc = [", paste(round(ci,3), collapse = " to "), "] - confidence interval", sep=""))
```

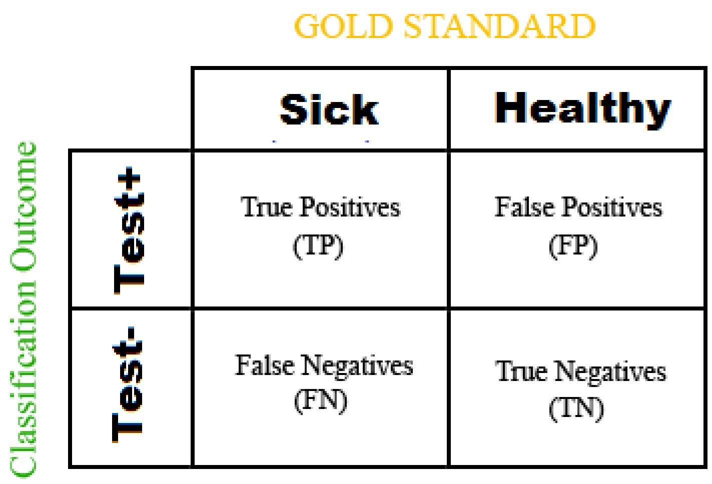
```
## [1] "CI calc = [-213.02 to -179.2] - confidence interval"
```

```
print(paste("CI pack = [", paste(round(tt$conf.int,3), collapse = " to "), "] - confidence interval", sep=""))
```

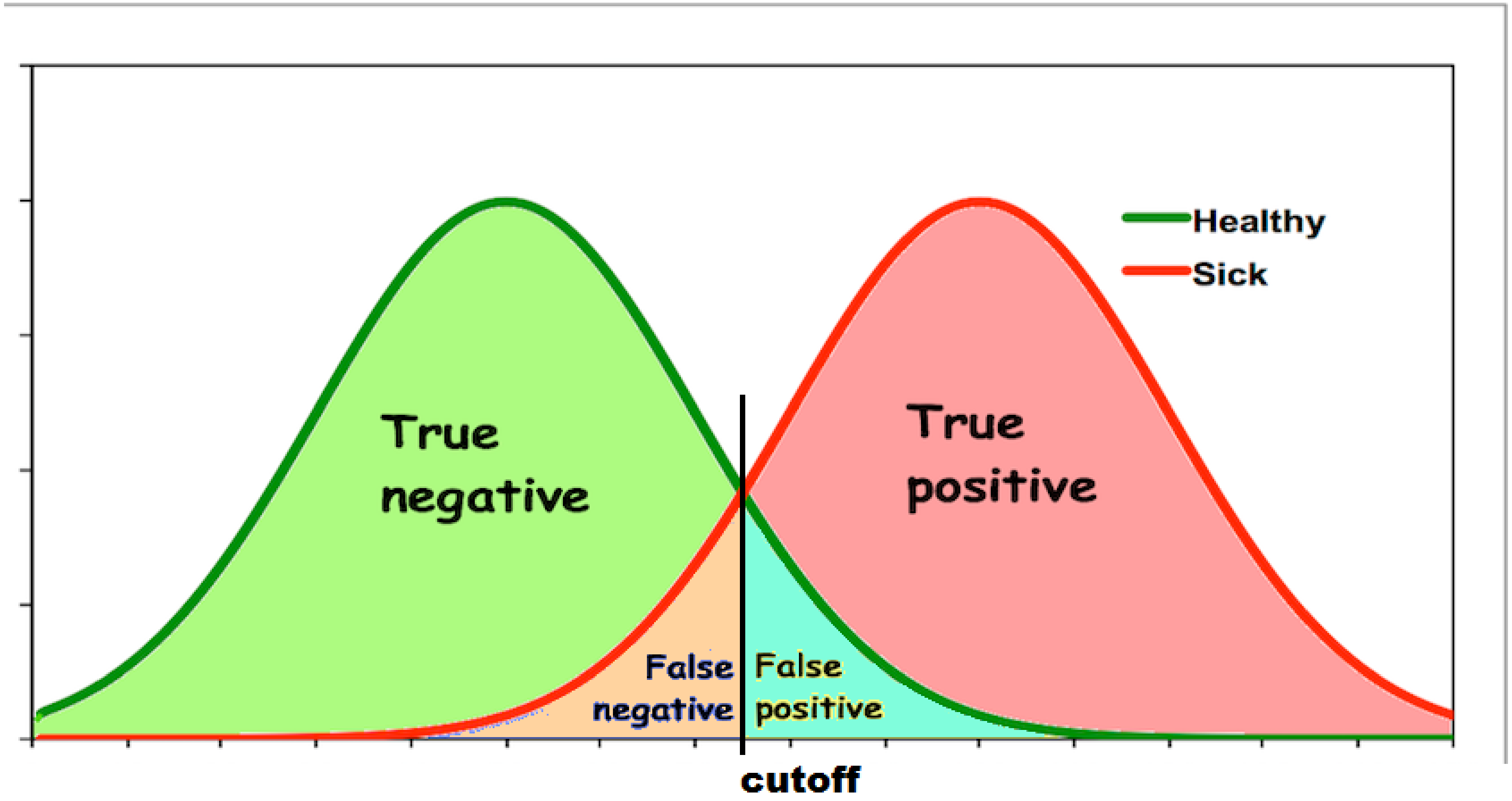
```
## [1] "CI pack = [-213.968 to -178.252] - confidence interval"
```

```
print_2densities(healthy, sick, title="healthy x sick", classes=c("healthy", "sick"))
```


healthy x sick
density
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
x
class
healthy
sick



		GOLD STANDARD	
		Sick	Healthy
Classification Outcome	Test+	True Positives (TP)	False Positives (FP)
	Test-	False Negatives (FN)	True Negatives (TN)



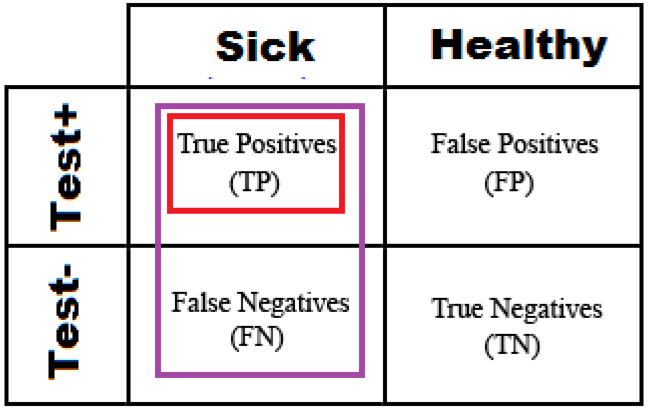
Conditional: truth table

https://en.wikipedia.org/wiki/Confusion_matrix (https://en.wikipedia.org/wiki/Confusion_matrix)

Sensitity

Sensitivity, recall, hit rate, or true positive rate (TPR)

$$Sensitivity = TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$



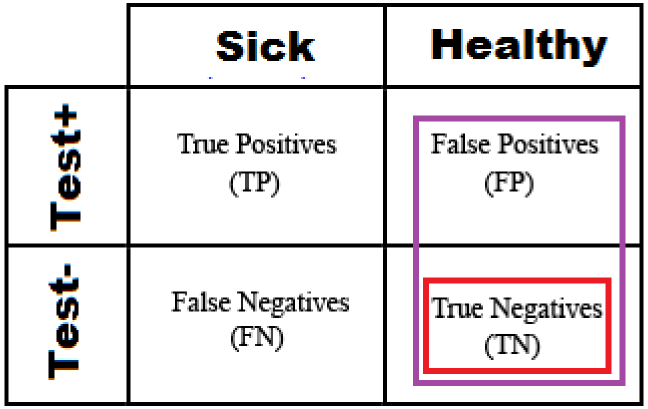
	Sick	Healthy
Test+	True Positives (TP)	False Positives (FP)
Test-	False Negatives (FN)	True Negatives (TN)

Conditional: sensitivity

Especifity

specificity or true negative rate (TNR)

$$Especifity = TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$$



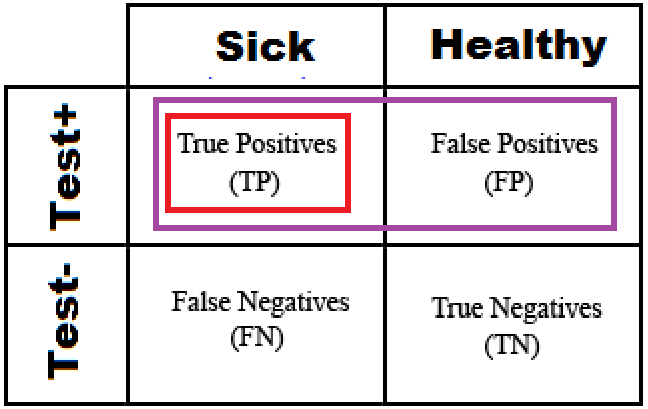
	Sick	Healthy
Test+	True Positives (TP)	False Positives (FP)
Test-	False Negatives (FN)	True Negatives (TN)

Conditional: specificity

Precision

precision or positive predictive value (PPV)

$$Precision = PPV = \frac{TP}{TP + FP}$$



	Sick	Healthy
Test+	True Positives (TP)	False Positives (FP)
Test-	False Negatives (FN)	True Negatives (TN)

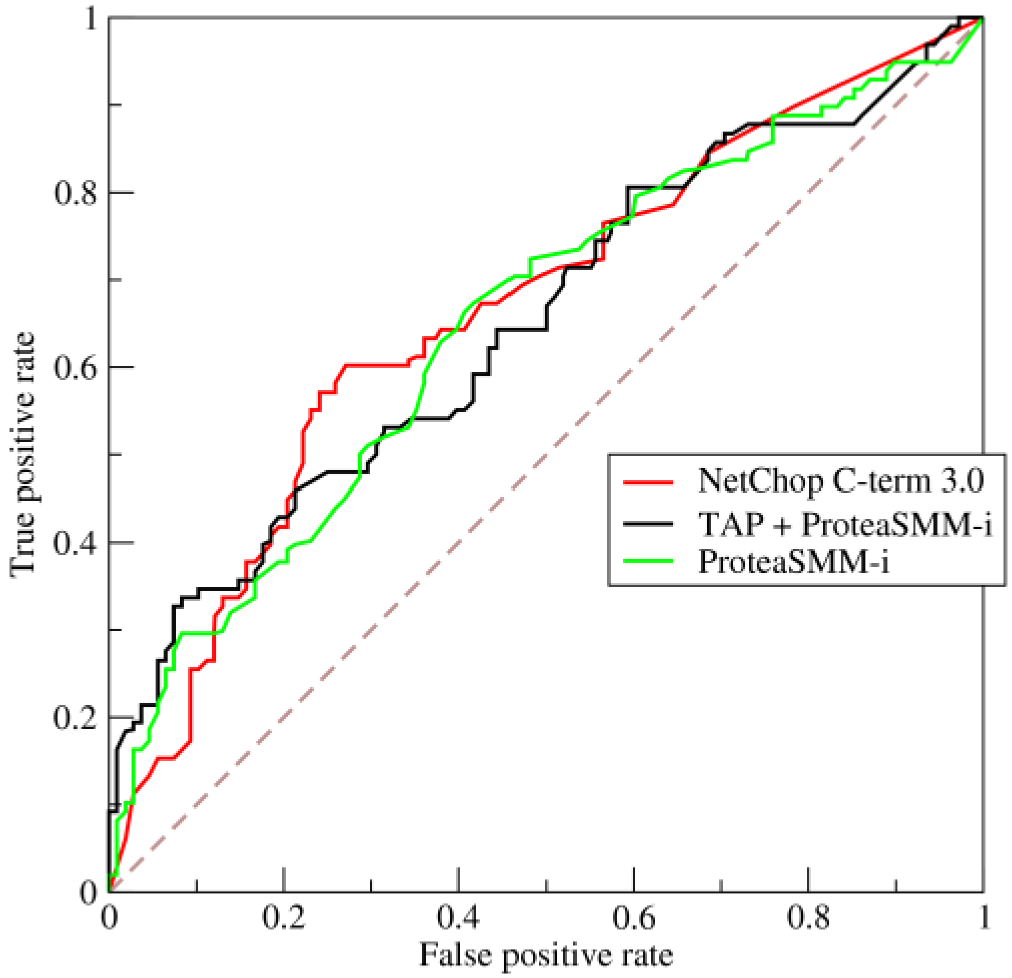
Conditional: Precision

ROC - Receiver Operating Curve

In statistics, a receiver operating characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection[1] in machine learning.

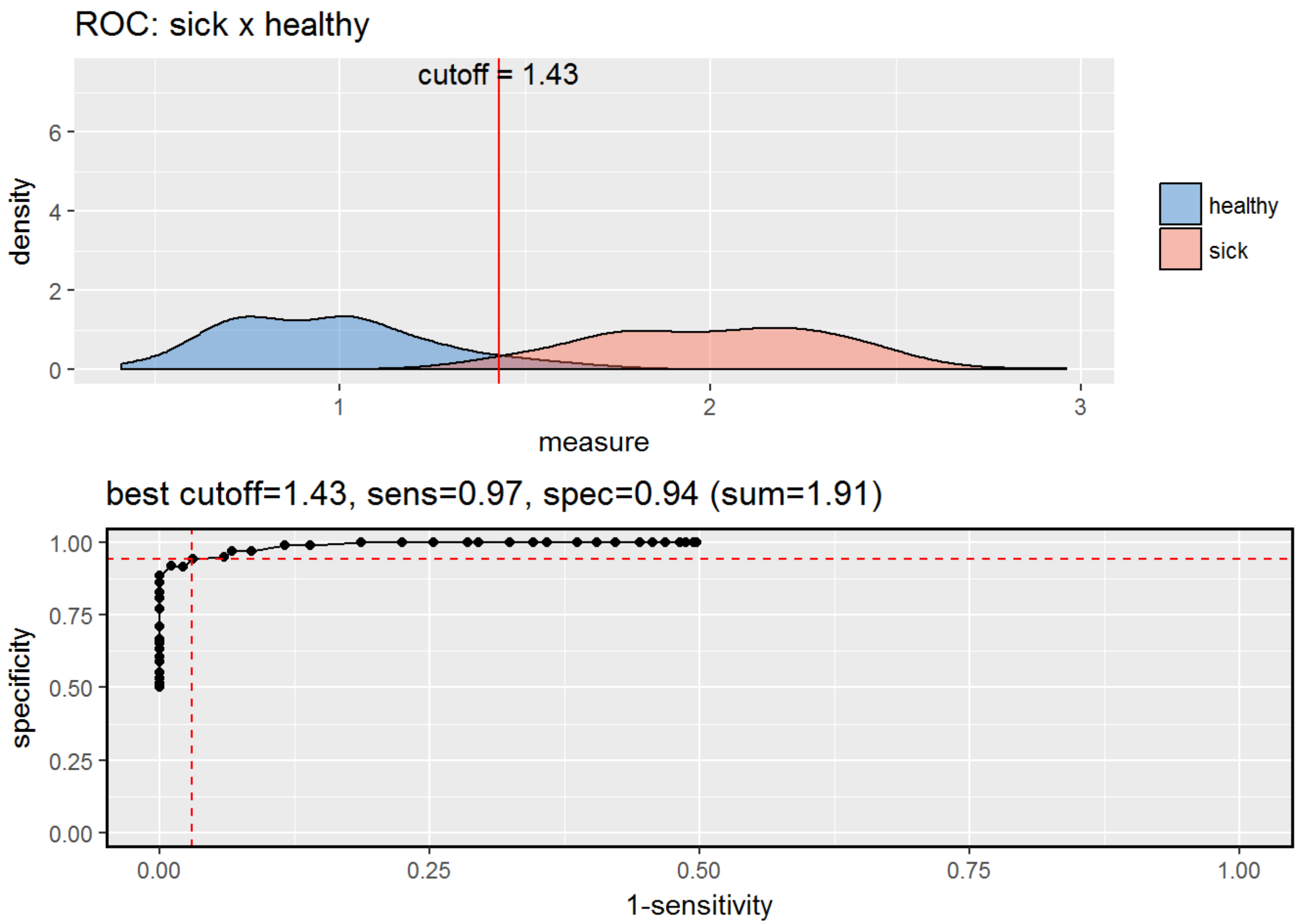
https://en.wikipedia.org/wiki/Receiver_operating_characteristic (https://en.wikipedia.org/wiki/Receiver_operating_characteristic)



```
# lista = contigency_table(healthy, sick)
# names(lista)

healthy = rnorm(100, mean=1, sd=.3)
sick    = rnorm(100, mean=2, sd=.3)

lista = doRoc(healthy, sick, "healthy", "sick", "healthy", "sick", xlab="measure", isLog=F, titleAux="")
names(lista)
```

```
## [1] "p1"         "p2"         "best_sens" "best_spec" "cutoff"
```

```
p1 = lista[["p1"]]
p2 = lista[["p2"]]

multiplot(p1,p2,layout=matrix(c(1,2),ncol=1, byrow=F))
```


ROC: sick x healthy
cutoff = 1.43
density
measure
healthy
sick
best cutoff=1.43, sens=0.97, spec=0.94 (sum=1.91)
specificity
1-sensitivity