



ROC and AUC in R with a Single Binary Predictor

John Muschelli

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health

Abstract

The abstract of the article.

Keywords: roc, auc, area under the curve, R.

1. Introduction

```
R> library(cranlogs)
R> library(ggplot2)
R> library(dplyr)
```

In many applications receiver operator characteristic (ROC) curves are used to show
In this tutorial, we will show the calculation of the AUC

2. Simple Example

Here we will create a simple scenario where there is a binary predictor X and a binary outcome Y .

```
R> x = c(rep(0, 52), rep(1, 32),
R+      rep(0, 35), rep(1, 50))
R> y = c(rep(0, 84), rep(1, 85))
R> tab = table(x, y)
R> tab
```

	y	
x	0	1
0	52	35
1	32	50

2.1. Mathematical Proof of AUC for Single Binary Predictor

As there are only two outcomes for X , we can expand the probability using the law of total probability:

$$P(X_1 > X_0) = P(X_1 > X_0 | X_1 = 1)P(X_1 = 1) + P(X_1 > X_0 | X_1 = 0)P(X_1 = 0) \quad (1)$$

$$= P(X_1 > X_0 | X_1 = 1)P(X_1 = 1) \quad (2)$$

where the second term of equation (1) is equal to zero because $X_0 \in \{0, 1\}$.

Here we see that the second term of equation (2) is the sensitivity:

$$\begin{aligned} P(X_1 = 1) &= P(X = 1 | Y = 1) \\ &= \frac{TP}{TP + FN} \\ &= \text{sensitivity} \end{aligned}$$

Here we show the first term of equation (2) is the specificity:

$$\begin{aligned} P(X_1 > X_0 | X_1 = 1) &= P(X_1 > X_0 | X_1 = 1, X_0 = 1)P(X_0 = 1) \\ &\quad + P(X_1 > X_0 | X_1 = 1, X_0 = 0)P(X_0 = 0) \\ &= P(X_1 > X_0 | X_1 = 1, X_0 = 0)P(X_0 = 0) \\ &= P(X_0 = 0) \\ &= P(X = 0 | Y = 0) \\ &= \frac{TN}{TN + FP} \\ &= \text{specificity} \end{aligned}$$

Therefore, we combine these two to show that equation (2) reduces to:

$$P(X_1 > X_0) = \text{specificity} * \text{sensitivity}$$

Therefore, the true AUC should be equal to:

```
R> sens = tab[2,2] / sum(tab[,2])
R> spec = tab[1,1] / sum(tab[,1])
R> true_auc = sens * spec
R> print(true_auc)
```

```
[1] 0.3641457
```

Reverse Labeling

```
R> flip_auc = (1 - sens) * (1 - spec)
R> print(flip_auc)
```

```
[1] 0.1568627
```

```
R> fpr = 1-spec
R> area_of_tri = 1/2 * sens * fpr
R> area_of_quad = sens * spec + 1/2 * spec * (1-sens)
R> auc = area_of_tri + area_of_quad
```

We can also show that if we use a simple sampling method, we can estimate this true AUC. Here, the function `est_auc` samples 10^6 random samples from X_1 and X_0 , then calculates $\hat{P}(X_1 > X_0)$:

```
R> est_auc = function(x, y) {
R+   x1 = x[y == 1]
R+   x0 = x[y == 0]
R+   n = 1000000
R+   c1 = sample(x1, size = n, replace = TRUE)
R+   c0 = sample(x0, size = n, replace = TRUE)
R+   mean(c1 > c0)
R+ }
R> sample_est_auc = est_auc(x, y)
R> sample_est_auc
```

```
[1] 0.364464
```

3. Current Implementations

3.1. R

caTools Package

The **caTools** package is one of the most popular packages in R, and has analysis for doing area under the curve:

```
R> library(caTools)
R> colAUC(x, y)
```

```
[,1]
0 vs. 1 0.6036415
```

ROCR Package

The **ROCR** package is one of the most popular packages for doing ROC analysis ([Sing, Sander, Beerenwinkel, and Lengauer 2005](#)). Using `prediction` and `performance` functions, we see that the estimated AUC is much higher than the true AUC:

```
R> library(ROCR)
```

```
Loading required package: gplots
```

```
Attaching package: 'gplots'
```

```
The following object is masked from 'package:stats':
```

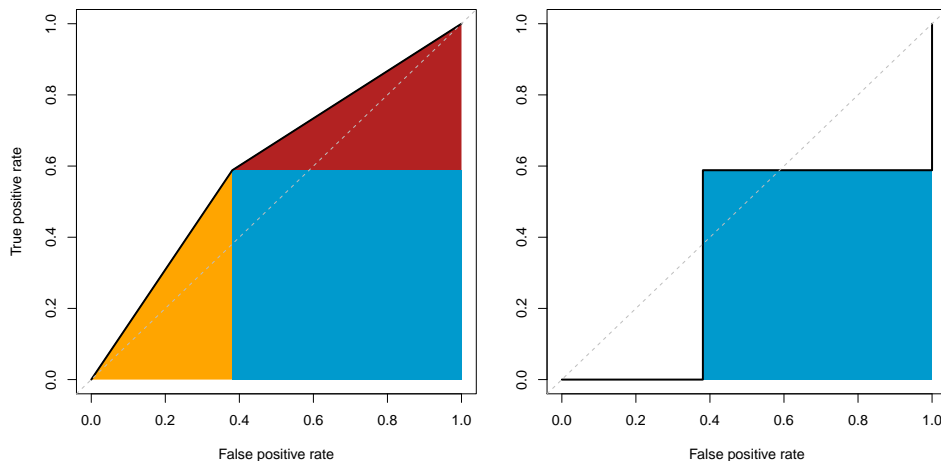
```
lowess
```

```
R> pred = prediction(x, y)
R> auc_est = performance(pred, "auc")
R> auc_est@y.values[[1]]
```

```
[1] 0.6036415
```

Looking at the plot for the ROC curve in ROCR, we can see why this may be:

```
R> par(mfrow = c(1, 2))
R> perf = performance(pred, "tpr", "fpr")
R> plot(perf)
R> abline(a = 0, b = 1)
R> plot(perf, type = "s")
R> abline(a = 0, b = 1)
```



Looking geometrically at the plot, we can see how

```
R> fpr = 1 - spec
R> area_of_left_tri = 1/2 * sens * fpr
R> area_of_top_tri = 1/2 * spec * (1 - sens)
R> false_auc = area_of_left_tri + true_auc + area_of_top_tri
R> false_auc
```

```
[1] 0.6036415
```

pROC Package

The **pROC** package is one of the most popular packages for doing ROC analysis ([Robin, Turck, Hainard, Tiberti, Lisacek, Sanchez, and Müller 2011](#)). Using `prediction` and `performance` functions, we see that the estimated AUC is much higher than the true AUC:

```
R> library(pROC)
```

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

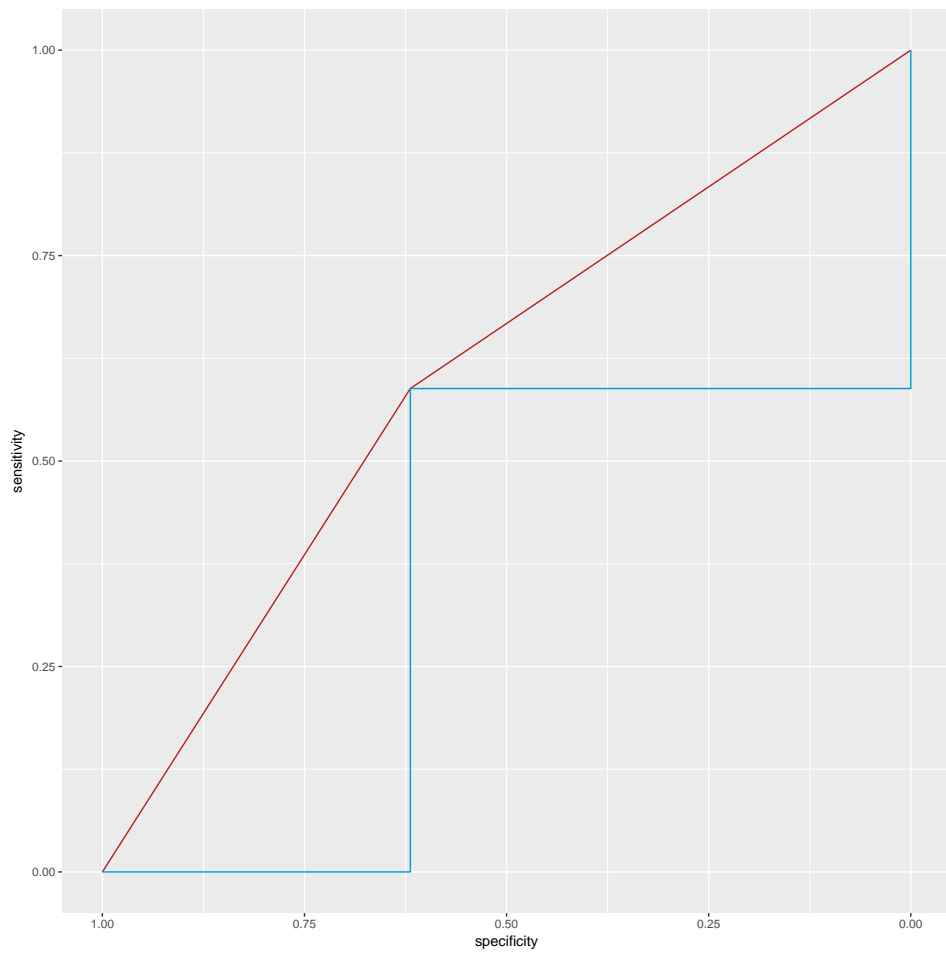
Attaching package: 'pROC'

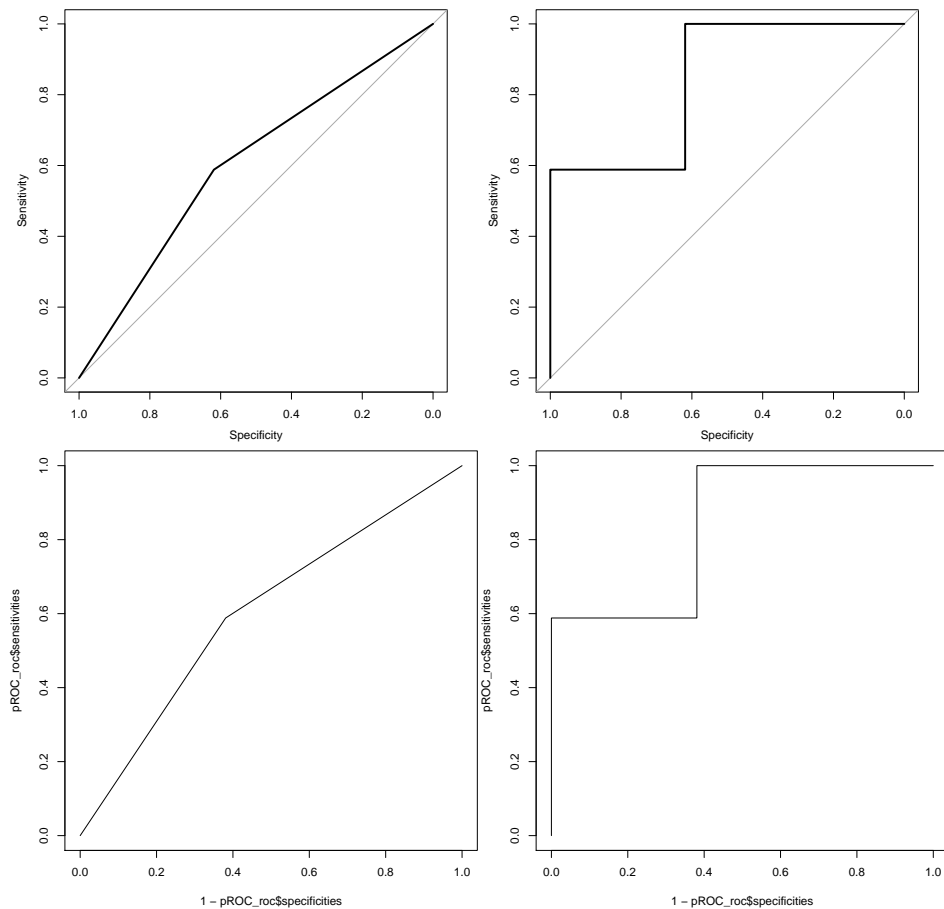
The following objects are masked from 'package:stats':

cov, smooth, var

```
R> pROC_roc = pROC::roc(predictor = x, response = y)
R> pROC_roc[["auc"]]
```

Area under the curve: 0.6036





fbroc Package

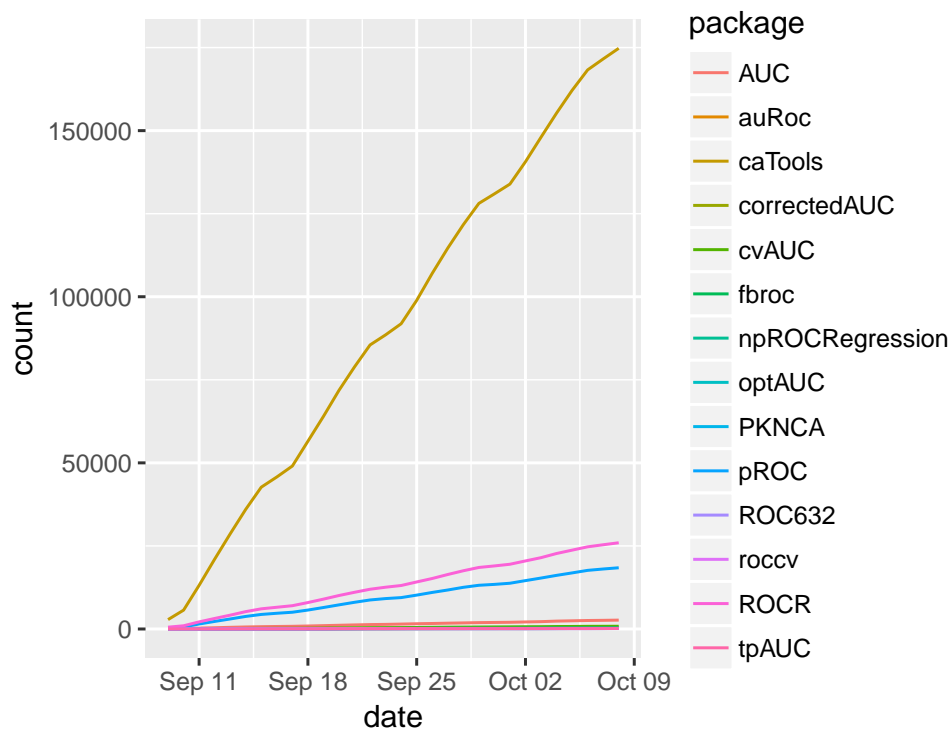
The **fbroc** package is one of the most popular packages for doing ROC analysis ([Peter 2016](#)). Using `prediction` and `performance` functions, we see that the estimated AUC is much higher than the true AUC:

```
R> library(fbroc)
R> fb_roc_default = boot.roc(x, as.logical(y), n.boot = 1000, tie.strategy = 2)
R> auc_def = perf(fb_roc_default, "auc")
R> auc_def[["Observed.Performance"]]

[1] 0.6036415

R> fb_roc_alternative = boot.roc(x, as.logical(y), n.boot = 1000, tie.strategy = 1)
R> auc_alt = perf(fb_roc_alternative, "auc")
R> auc_alt[["Observed.Performance"]]

[1] 0.6036415
```



3.2. Stata

```
R> roctab x y
```

```
. roctab x y
```

Obs	ROC Area	Std. Err.	-Asymptotic Normal-- [95% Conf. Interval]	
169	0.6037	0.0379	0.52952	0.67793

References

- Peter E (2016). *fbroc: Fast Algorithms to Bootstrap Receiver Operating Characteristics Curves*. R package version 0.4.0, URL <https://CRAN.R-project.org/package=fbroc>.
- Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M (2011). “pROC: an open-source package for R and S+ to analyze and compare ROC curves.” *BMC Bioinformatics*, **12**, 77.
- Sing T, Sander O, Beerenwinkel N, Lengauer T (2005). “ROCR: visualizing classifier performance in R.” *Bioinformatics*, **21**(20), 7881. URL <http://rocr.bioinf.mpi-sb.mpg.de>.

Affiliation:

John Muschelli

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health

615 N Wolfe St Baltimore, MD 21205

E-mail: jmuschel@jhsph.edu

URL: <http://johnmuscHELLi>.