Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

ROC and AUC with a 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

In many applications, receiver operator characteristic (ROC) curves are used to show how a predictor compares to the true outcome. One of the large advantages of ROC analysis is that it is threshold-agnostic; it shows the performance of a predictor without a specific threshold and also gives a criteria to choose an optimal threshold based on a certain cost function. Typically, an ROC analysis shows how sensitivity changes with varying specificity.

Many predictors, especially medical tests, result in a binary decision; a value is higher than a pre-determined threshold or not. Similarly, some are simply categorical or discrete; for example, blood pressure may be low, normal, or high. These are useful indicators of presence disease, which is a primary outcome of interest. The predictive capabilities of the variable is summarized many times in the area under the curve (AUC). Additionally, partial ROC (pROC) analysis keeps a specificity fixed and determines the optimal sensitivity or the partial AUC (pAUC).

One can assume the binary predictor is generated from a continuous distribution that has been thresholded. Therefore, the sensitivity of this thresholded predictor actually represents one point on the ROC curve of the true underlying continuous distribution. Therefore the ROC curve of a binary predictor is not really appropriate, but should be represented by a single point on the curve. But alas, ROC and AUC analysis is done on binary predictors and used to inform if one variable is more predictive than the other (E, C, K, and et al 2018, TV, GH, JA, S, K, and GY (2017)). For example, these cases

A more appropriate comparison of a continuous predictor and the binary predictor may be to

compare the sensitivity and specificity (or overall accuracy) of the continuous predictor given the optimal threshold versus that of the binary predictor.

Fawcett (2006) describes (in Fig. 6) how ties are handled in a predictor. Ties are distinctly relevant for discrete and binary predictors or models that predict a discrete number of values, where many observations can have the same value/risk score. When drawing the ROC curve, one can assume that all the ties do not correctly classify the outcome (Fawcett called the "pessimistic" approach) or that all the ties do correclty classify the outcome (called the "optimistic" approach). But Fawcett notes:

> Any mixed ordering of the instances will give a different set of step segments within the rectangle formed by these two extremes. However, the ROC curve should represent the *expected* performance of the classifier, which, lacking any other information, is the average of the pessimistic and optimistic segments.

This "expected" performance directly applies to the assignment of a half probability of success when the data are tied, which is implied by the "trapezoidal rule" from Hanley and McNeil (1982). This half probability is linked to how ties are treated in the Wilcoxon rank sum test. As much of the theory of ROC curve testing, and therefore testing of differences in AUC, is based on the theory of the Wilcoxon rank-sum test, this treatment of ties is relevant to statistical inference.

We show the commonly used software for calculating AUC may be misleading for binary or categorical predictors depending on the definition of the AUC.

2. Mathematical Proof of AUC for Single Binary Predictor

Let us assume we have a binary predictor X and a binary outcome Y, such that X and Y only take the values 0 and 1. Let X_i be the values of $X|Y = i$, where $i \in \{0, 1\}$.

Fawcett goes on to state: $>$ AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

In other words, we AUC $= P(X_1 > X_0)$. As there are only two outcomes for X, we can expand this probability using the law of total probability:

$$\begin{aligned} P(X_1 > X_0) &= P(X_1 > X_0|X_1 = 1)P(X_1 = 1) \\ &\quad + P(X_1 > X_0|X_1 = 0)P(X_1 = 0) \qquad (1) \\ &= P(X_1 > X_0|X_1 = 1)P(X_1 = 1) \qquad (2) \end{aligned}$$

as $P(X_1 > X_0|X_1 = 0) = 0$ because $X_0 \in \{0, 1\}$. We see that $P(X_1 = 1)$ in equation (2) is the sensitivity:

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

and that $P(X_1 > X_0|X_1 = 1)$ in equation (2) is the specificity:

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

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

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

2.1. Simple Example

Let's assume X and Y have the following joint distribution:

	0	1
0	52	35
1	32	50

Therefore, the AUC should be equal to $\frac{50}{85} \times \frac{52}{84}$, which equals:

```
[1] 0.3641457
```

As this is the strict definition of AUC, let us call this $\text{AUC}_{\text{definition}}$.

Note, if we reverse the labels, then the sensitivity and the specificity are estimated by 1 minus that measure, or $\frac{35}{85} \times \frac{32}{84}$, which is equal to:

```
R> flip.auc = (1 - sens) * (1 - spec)
R> print(flip.auc)

[1] 0.1568627
```

Thus, as this AUC is less than the original labeling, we would choose that with the original labeling.

If we change the definition of AUC slightly while accounting for ties, $\text{AUC}_{\text{w/ties}}$ to

$$\text{AUC}_{\text{w/ties}} = P(X_1 > X_0) + \frac{1}{2}P(X_1 = X_0)$$

we see that we would calculate the AUC by $\text{AUC}_{\text{definition}} + \frac{1}{2}\left(\frac{50+52}{169}\right)$, which is equal to:

```
[1] 0.6036415
```

We will explore the estimated ROC curve and AUC from the implentations in the follwing `R` packages: **ROCR** (Sing, Sander, Beerenwinkel, and Lengauer 2005), **caTools**, **pROC** (Robin, Turck, Hainard, Tiberti, Lisacek, Sanchez, and Müller 2011), and **fbroc** (Peter 2016). We will also show these agree with the Python implementation in `sklearn.metrics` (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay 2011) and the Stata function `roctab`. We note that these functions all count half the probability of ties, which raises the AUC. We will present a geometric discussion of the ROC as well.

3. Current Implementations

3.1. R

The **caTools** package calculates AUC using the `colAUC` function:

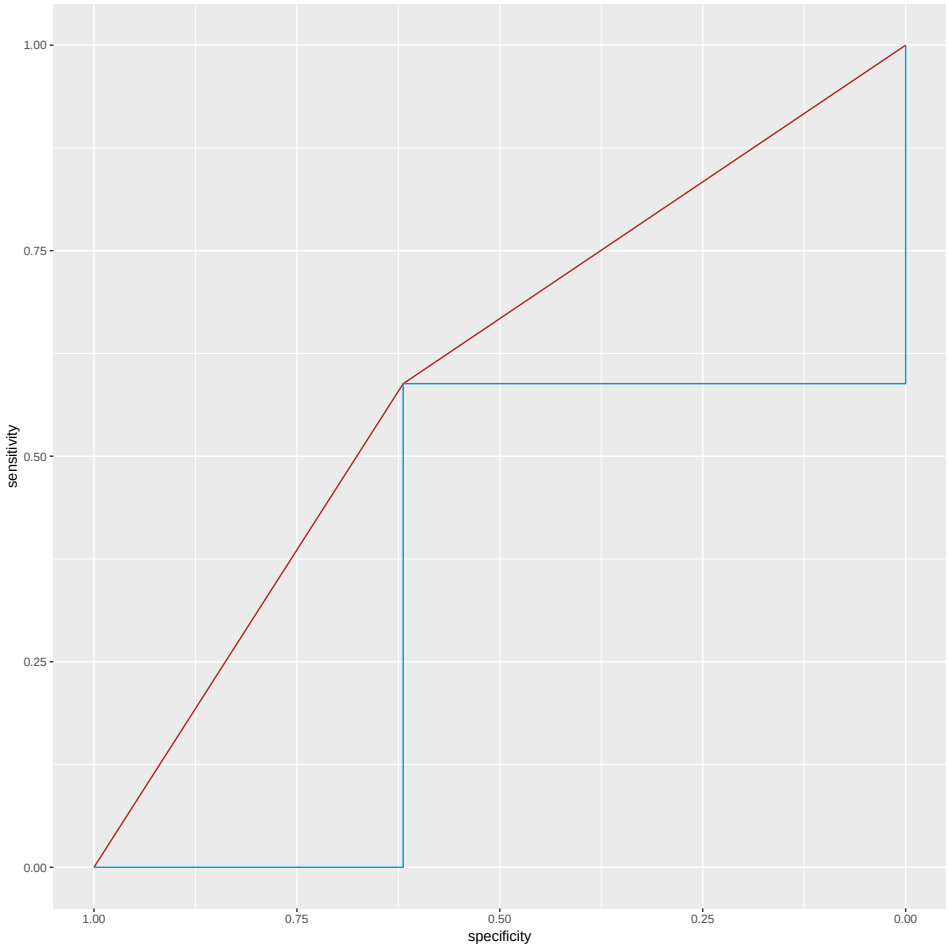
```
R> library(caTools)
R> colAUC(x, y)

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

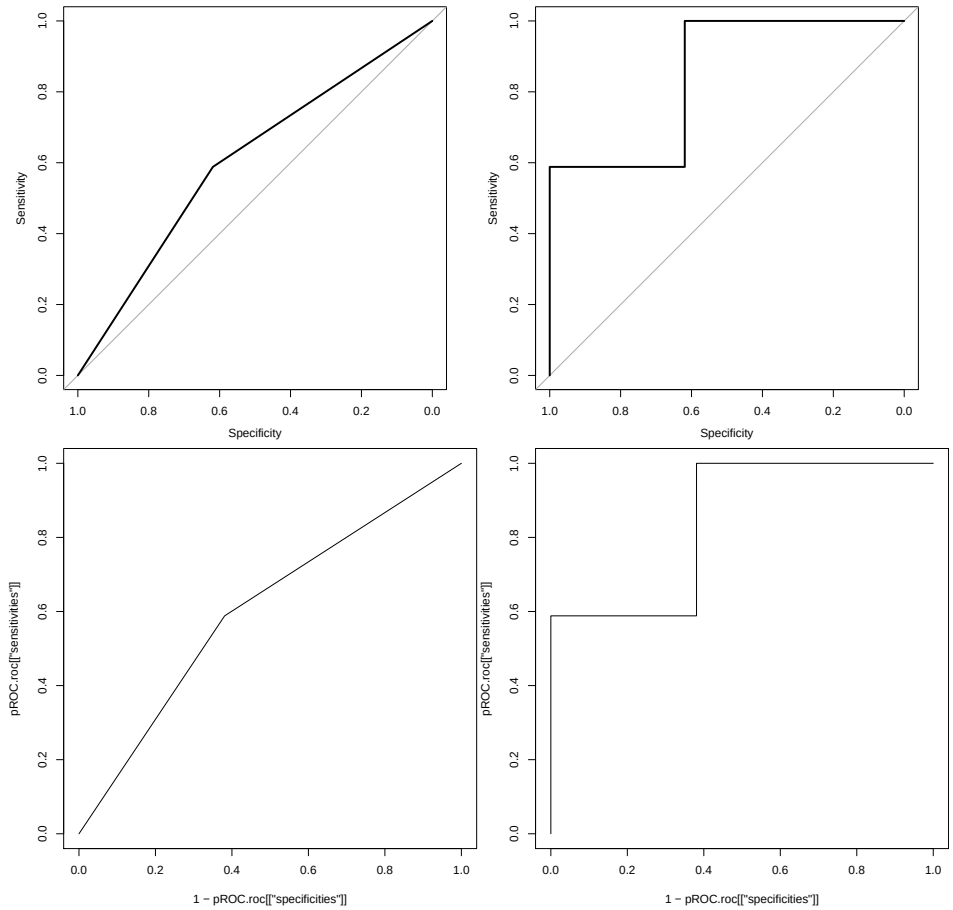
In **ROCR**, AUC is calculated from a `performance` object, which takes in a `prediction` object:

```
R> library(ROCR)
R> yvalues = function(perf) {
R+   unlist(slot(perf, "y.values"))
R+ }
R> xvalues = function(perf) {
R+   unlist(slot(perf, "x.values"))
R+ }
R>
R> pred = prediction(x, y)
R> auc.est = performance(pred, "auc")
R> yvalues(auc.est)

[1] 0.6036415
```

The **pROC** package calculates AUC using the `roc` function:

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

Area under the curve: 0.6036
```


1.00
0.75
0.50
0.25
0.00
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity



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> left.tri = 1/2 * sens * fpr
R> right.tri = 1/2 * spec * (1 - sens)
R> false.auc = left.tri + auc.defn + right.tri
R> false.auc
```

```
[1] 0.6036415
```

pROC Package

http://blog.revolutionanalytics.com/2016/11/calculating-auc.html

fbroc Package

The **fbroc** package is one of the most popular packages for doing ROC analysis (Peter 2016). Using the `fbroc::boot.roc` and `fbroc::perf` functions, we have:

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

[1] 0.6036415

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

[1] 0.6036415
```

\begin{CodeChunk}
\begin{CodeInput} R> python_figure = "python_roc.png" R> # Taken from R> # https://qiita.com/bmj0114/items/460424c110a8ce22d945 R> sk = import("sklearn.metrics") R> plt = import("matplotlib.pyplot") R> py_roc_curve = sk$roc_curve(y_score = x, y_true = y)R > names(py_roc_curve) = c("fpr","tpr","thresholds")R > roc_auc = sk$auc(py_roc_curve$fpr, py_roc_curve$tpr) R> R> plt$figure() \end{CodeInput}

```
Figure(640x480)
```

\begin{CodeInput} R> plt$plot(py_roc_curve$fpr, py_roc_curve$tpr, color='darkorange', lw=1, label= sprintf('ROC curve (area = %0.3f)', roc_auc)) \end{CodeInput}

```
[[1]]
Line2D(ROC curve (area = 0.604))

R> plt$plot(c(0, 1), c(0, 1), color='navy', lw=1, linestyle='--')

[[1]]
Line2D(_line1)

R> plt$xlim(c(0.0, 1.0))

[[1]]
[1] 0

[[2]]
[1] 1
```

```
R> plt$ylim(c(0.0, 1.05))
```

```
[[1]]
[1] 0

[[2]]
[1] 1.05
```

```
R> plt$xlabel('False Positive Rate')
```

```
Text(0.5,0,'False Positive Rate')
```

```
R> plt$ylabel('True Positive Rate')
```

```
Text(0,0.5,'True Positive Rate')
```

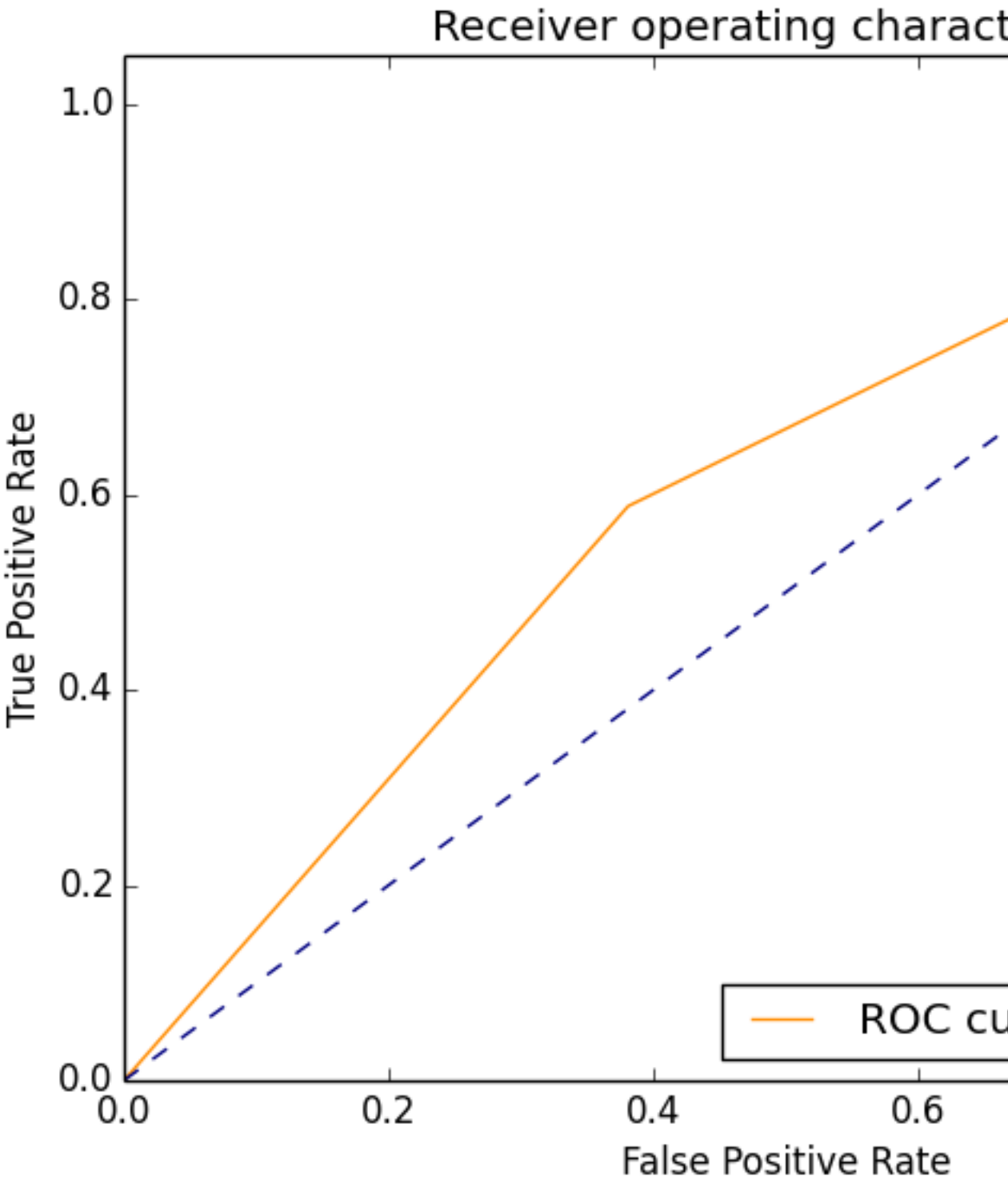
```
R> plt$title('Receiver operating characteristic')
```

```
Text(0.5,1,'Receiver operating characteristic')
```

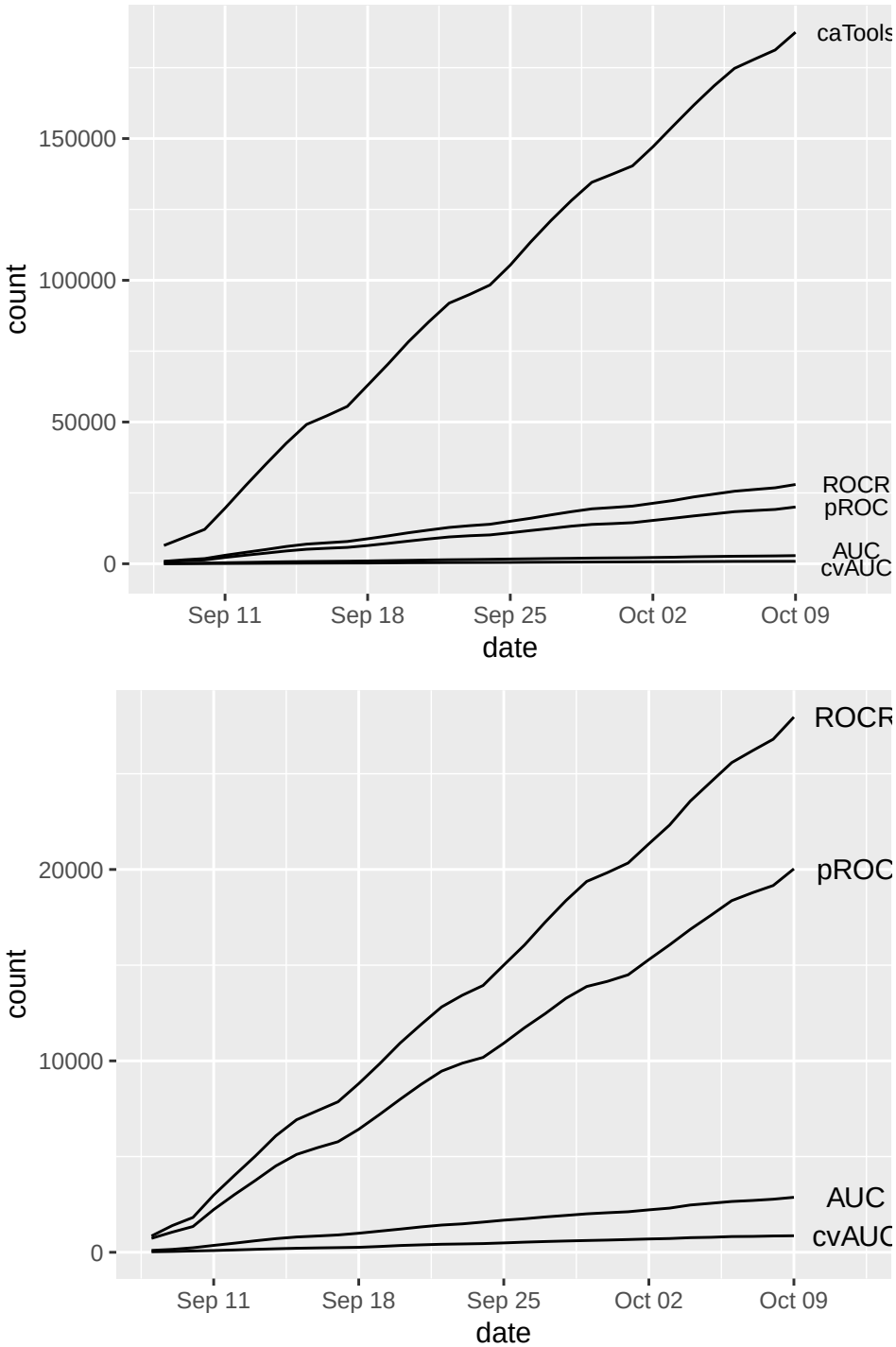
\begin{CodeInput} R> plt$legend(loc="lower right") \end{CodeInput}

```
Legend
```

\begin{CodeInput} R> plt$savefig(python_figure) \end{CodeInput} \end{CodeChunk}
\begin{CodeChunk}
\begin{CodeInput} R> knitr::include_graphics(python_figure) \end{CodeInput}



\end{CodeChunk}



3.2. Stata

```
R> roctab x y
```

```
 . roctab x y
```

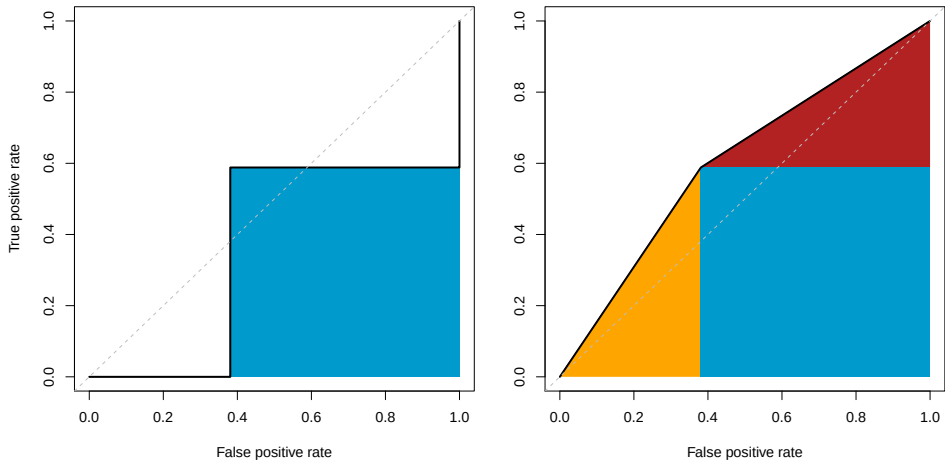


Figure 1: hey

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

We can also show that if we use simple Monte Carlo sampling, we can estimate this true AUC, based on the definition above. Here, the function `est.auc` samples 10^6 random samples from X_1 and X_0, then calculates $\hat{P}(X_1 > X_0)$:

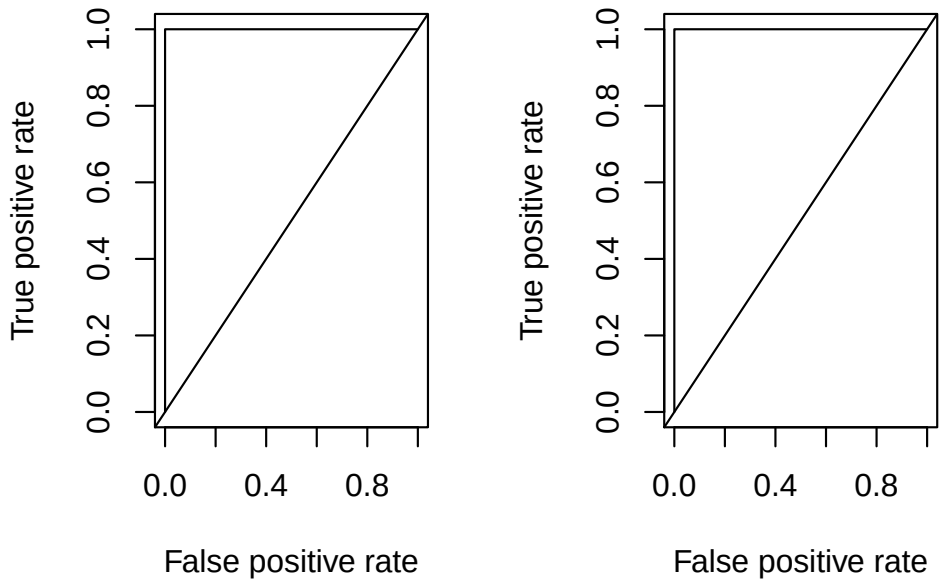
```
R> est.auc = function(x, y, n = 1000000) {
R+   x1 = x[y == 1] # sample x | y = 1
R+   x0 = x[y == 0] # sample x | y = 0
R+   c1 = sample(x1, size = n, replace = TRUE)
R+   c0 = sample(x0, size = n, replace = TRUE)
R+   auc.defn = mean(c1 > c0) # compare
R+   auc.wties = auc.defn + 1/2 * mean(c1 == c0) # compare
R+   return(c(auc.defn = auc.defn,
R+             auc.wties = auc.wties))
R+ }
R> sample.estauc = est.auc(x, y)
R> sample.estauc

 auc.defn auc.wties
0.3646780 0.6039805
```

One should note, that

3.3. Fawcett Example

```
R> faw = data.frame(y = c(rep(TRUE, 6), rep(FALSE, 4)),
R+                  x = c(0.99999, 0.99999, 0.99993,
```

```
R+                          0.99986, 0.99964, 0.99955,
R+                          0.68139, 0.50961, 0.48880, 0.44951))
R> faw = faw %>% mutate(hyp = x > 0.5)
R> pred = prediction(predictions = faw[, "x"], labels = faw[, "y"])
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)
```




```
R> auc.estimated = performance(pred, "auc")
R> yvalues(auc.estimated)

[1] 1

R> est.auc(x = faw[, "x"], y = faw[, "y"])

 auc.defn auc.wties
        1         1
```

References

E M, C M, K S, et al (2018). "Diagnostic criteria of ulcerative pyoderma gangrenosum: A delphi consensus of international experts." *JAMA Dermatology*, **154**(4), 461–466. doi:10.1001/jamadermatol.2017.5980. /data/journals/derm/936924/jamadermatology_maverakis_2018_cs_170003.pdf, URL +http://dx.doi.org/10.1001/jamadermatol.2017.5980.

Fawcett T (2006). "An introduction to ROC analysis." *Pattern recognition letters*, **27**(8), 861–874.

Hanley JA, McNeil BJ (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, **143**(1), 29–36.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, **12**, 2825–2830.

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.

TV L, GH B, JA C, S S, K K, GY O (2017). "A revised approach for the detection of sight-threatening diabetic macular edema." *JAMA Ophthalmology*, **135**(1), 62–68. doi:10.1001/jamaophthalmol.2016.4772. /data/journals/ophth/935943/jamaophthalmology_litvin_2016_oi_160098.pdf, URL +http://dx.doi.org/10.1001/jamaophthalmol.2016.4772.

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.com