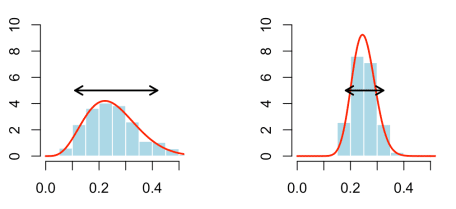
Each time we have a case study in my actuarial courses (with real data), students are surprised to have hard time getting a “good” model, and they are always surprised to have a low [AUC](https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it), when trying to model the probability to claim a loss, to die, to fraud, etc. And each time, I keep saying, “yes, I know, and that’s what we expect because there a lot of ‘randomness’ in insurance”. To be more specific, I decided to run some simulations, and to compute AUCs to see what’s going on. And because I don’t want to waste time fitting models, we will assume that we have each time a perfect model. So I want to show that the upper bound of the AUC is actually quite low ! So it’s not a modeling issue, it is a fondamental issue in insurance !

By ‘perfect model’ I mean the following : denotes the heterogeneity factor, because people are different. We would love to get . Unfortunately,   is unobservable ! So we use covariates (like the age of the driver of the car in motor insurance, or of the policyholder in life insurance, etc). Thus, we have data ‘s and we use them to train a model, in order to approximate . And then, we check if our model is good (or not) using the ROC curve, obtained from confusion matrices, comparing ‘s and ‘s where when exceeds a given threshold. Here, I will not try to construct models. I will predict each time the true underlying probability exceeds a threshold ! The point is that it’s possible to claim a loss () even if the probability is 3% (and most of the time ), and to not claim one () even if the probability is 97% (and most of the time ). That’s the idea with randomness, right ?

So, here denote the probabilities to claim a loss, to die, to fraud, etc. There is heterogeneity here, and this heterogenity can be small, or large. Consider the graph below, to illustrate,



In both cases, there is, on average, 25% chance to claim a loss. But on the left, there is more heterogeneity, more dispersion. To illustrate, I used the arrow, which is a classical 90% interval : 90% of the individuals have a probability to claim a loss in that interval. (here 10%-40%), 5% are below 10% (low risk), and 5% are above 40% (high risk). Later on, we will say that we have 25% on average, with a dispersion of 30% (40% minus 10%). On the right, it’s more 25% on average, with a dispersion of of 15%. What I call dispersion is the difference between the 95% and the 5% quantiles.

Consider now some dataset, with Bernoulli variables , drawn with those probabilities . Then, let us assume that we are able to get a perfect model : I do not estimate a model based on some covariates, here, I assume that I know perfectly the probability (which is true, because I did generate those data). More specifically, to generate a vector of probabilities, here I use a Beta distribution with a given mean, and a given variance (to capture the heterogeneity I mentioned above)

|  |  |
| --- | --- |
| 1  2  3 | a=m\*(m\*(1-m)/v-1)  b=(1-m)\*(m\*(1-m)/v-1)  p=**rbeta**(n,a,b) |

from those probabilities, I generate occurences of claims, or deaths,

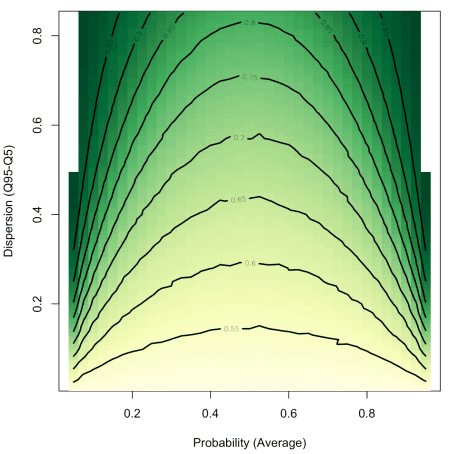
|  |  |
| --- | --- |
| 1 | Y=**rbinom**(n,size = 1,prob = p) |

Then, I compute the [AUC](https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it) of my “perfect” model,

|  |  |
| --- | --- |
| 1 | auc.tmp=performance(prediction(p,Y),"auc") |

And then, I will generate many samples, to compute the average value of the AUC. And actually, we can do that for many values of the mean and the variance of the Beta distribution. Here is the code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | **library**(ROCR)  n=1000  ns=200  ab\_beta = **function**(m,inter){  a=**uniroot**(**function**(a) **qbeta**(.95,a,a/m-a)-**qbeta**(.05,a,a/m-a)-inter,  interval=**c**(.0000001,1000000))$root  b=a/m-a  **return**(**c**(a,b))  }  Sim\_AUC\_mean\_inter=**function**(m=.5,i=.05){  V\_auc=**rep**(NA,ns)  b=-1  essai = **try**(ab&lt;-ab\_beta(m,i),TRUE) **if**(**inherits**(essai,what="try-error")) a=-1 **if**(!**inherits**(essai,what="try-error")){ a=ab[1] b=ab[2] } **if**((a&gt;=0)&amp;(b&gt;=0)){  **for**(s **in** 1:ns){  p=**rbeta**(n,a,b)  Y=**rbinom**(n,size = 1,prob = p)  auc.tmp=performance(prediction(p,Y),"auc")  V\_auc[s]=**as.numeric**(auc.tmp@y.values)}  L=**list**(moy\_beta=m,  var\_beat=v,  q05=**qbeta**(.05,a,b),  q95=**qbeta**(.95,a,b),  moy\_AUC=**mean**(V\_auc),  sd\_AUC=**sd**(V\_auc),  q05\_AUC=**quantile**(V\_auc,.05),  q95\_AUC=**quantile**(V\_auc,.95))  **return**(L)}  **if**((a&lt;0)|(b&lt;0)){**return**(**list**(moy\_AUC=NA))}}  Vm=**seq**(.025,.975,**by**=.025)  Vi=**seq**(.01,.5,**by**=.01)  V=**outer**(X = Vm,Y = Vi, **Vectorize**(**function**(x,y)  Sim\_AUC\_mean\_inter(x,y)$moy\_AUC))  **library**("RColorBrewer")  **image**(Vm,Vi,V,  xlab="Probability (Average)",  ylab="Dispersion (Q95-Q5)",  **col**=  **colorRampPalette**(brewer.pal(n = 9, name = "YlGn"))(101))  **contour**(Vm,Vi,V,add=TRUE,lwd=2) |

****

On the *x*-axis, we have the average probability to claim a loss. Of course, there is a symmetry here. And on the *y*-axis, we have the dispersion : the lower, the less heterogeneity in the portfolio. For instance, with a 30% chance to claim a loss on average, and 20% dispersion (meaning that in the portfolio, 90% of the insured have between 20% and 40% chance to claim a loss, or 15% and 35% chance), we have on average a 60% AUC. With a perfect model ! So with only a few covariates, having 55% should be great !

My point here is that with a low dispersion, we cannot expect to have a great AUC (again, even with a perfect model). In motor insurance, from my experience, 90% of the insured are between 3% chance and 20% chance to claim a loss ! That’s less than 20% dispersion ! and in that case, even if the (average) probability is rather small, it is very difficult to expect an AUC above 60% or 65% !