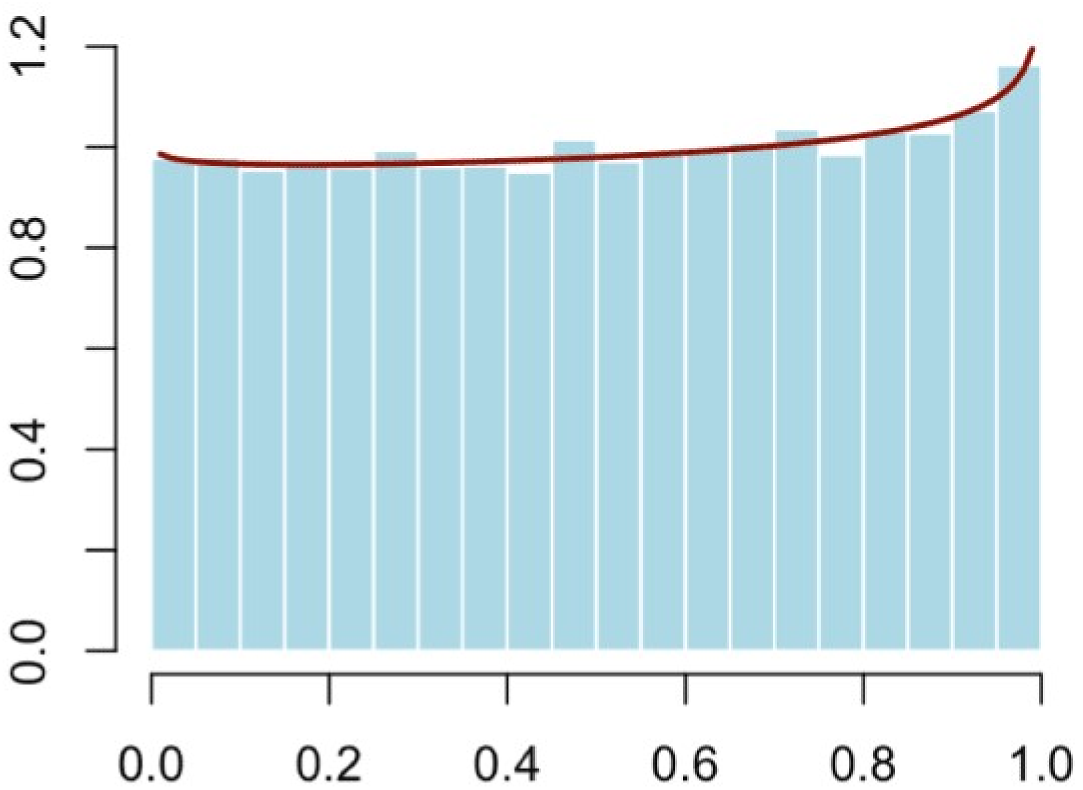
In statistics, Kolmogorov–Smirnov test is a popular procedure to test, from a sample $\{x_1,\cdots,x_n\}$ is drawn from a distribution $F$, or usually $F_{\theta_0}$, where $F_{\theta}$ is some parametric distribution. For instance, we can test $H_0:X_i\sim\mathcal{N(0,1)}$ (where $\theta_0=(\mu_0,\sigma_0^2)=(0,1)$) using that test. More specifically, I wanted to discuss today $p$-values. Given $n$ let us draw $\mathcal{N}(0,1)$ samples of size $n$, and compute the $p$-values of Kolmogorov–Smirnov tests

```
n=300
p = rep(NA,1e5)
for(s in 1:1e5){
X = rnorm(n,0,1)
p[s] = ks.test(X,"pnorm",0,1)$p.value
}
```

We can visualise the distribution of the $p$-values below (I added some Beta distribution fit here)

```
library(fitdistrplus)
fit.dist = fitdist(p,"beta")
hist(p,probability = TRUE,main="",xlab="",ylab="")
vu = seq(0,1,by=.01)
vv = dbeta(vu,shape1 = fit.dist$estimate[1], shape2 =
fit.dist$estimate[2])
lines(vu,vv,col="dark red", lwd=2)
```




It looks like it is quite uniform (theoretically, the $p$-value is uniform). More specifically, the $p$-value was lower than 5% in 5% of the samples [***note**: here I compute* ‘mean(p<=.05)’ *but I have some trouble with the* ‘<‘ *and* ‘>’ *symbols, as always*]

```
mean(p<=.05)
[1] 0.0479
```

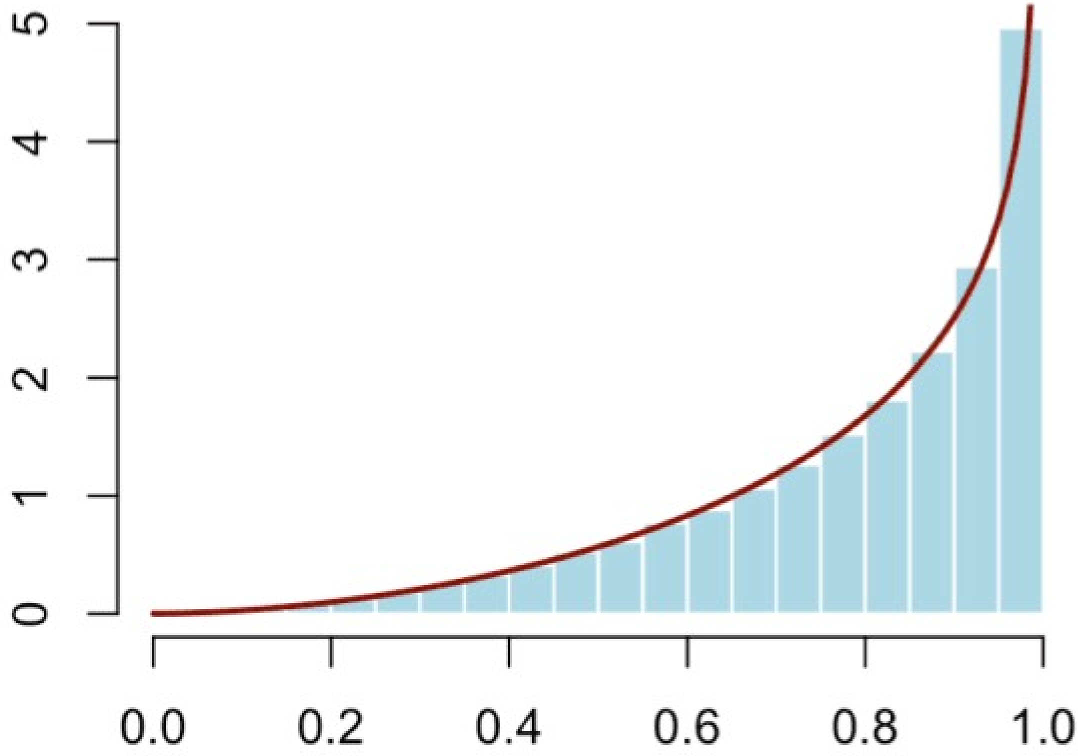
i.e. we wrongly reject $H_0:X_i\sim\mathcal{N(0,1)}$ is 5% of the samples.

As discussed previously on the blog, in many cases, we do care about the distribution, and not really the parameters, so we wish to test something like $H_0:X_i\sim\mathcal{N(\mu,\sigma^2)}$, for some $\mu$ and $\sigma^2$. Therefore, a natural idea can be to test $H_0:X_i\sim\mathcal{N(\hat\mu,\hat\sigma^2)}$, for some estimates of $\mu$ and $\sigma^2$. That’s the idea of Lilliefors test. More specifically, Lilliefors test suggests to use , Kolmogorov–Smirnov statistics, but corrects the $p$-value. Indeed, if we draw many samples, and use Kolmogorov–Smirnov statistics and its classical $p$-value to test for $H_0:X_i\sim\mathcal{N(\hat\mu,\hat\sigma^2)}$,

```
n=300
p = rep(NA,1e5)
for(s in 1:1e5){
X = rnorm(n,0,1)
p[s] = ks.test(X,"pnorm",mean(X),sd(X))$p.value
}
```

we see clearly that the distribution of $p$-values is no longer uniform

```
fit.dist = fitdist(p,"beta")
hist(p,probability = TRUE,main="",xlab="",ylab="")
vu = seq(0,1,by=.01)
vv = dbeta(vu,shape1 = fit.dist$estimate[1], shape2 =
fit.dist$estimate[2])
lines(vu,vv,col="dark red", lwd=2)
```




More specifically, if \(x_i\)'s are actually drawn from some Gaussian distribution, there are no chance to reject \(H_0\), the \(p\)-value being almost never below 5%

```
mean(p<=.05)
[1] 0.00012
```

Usually, to interpret that result, the heuristics is that \(\hat\mu\) and \(\hat\sigma^2\) are both based on the sample, while previously \(0\) and \(1\) where based on some prior knowledge. Somehow, it reminded me on the classical problem when mention when we introduce cross-validation, which is Goodhart's law

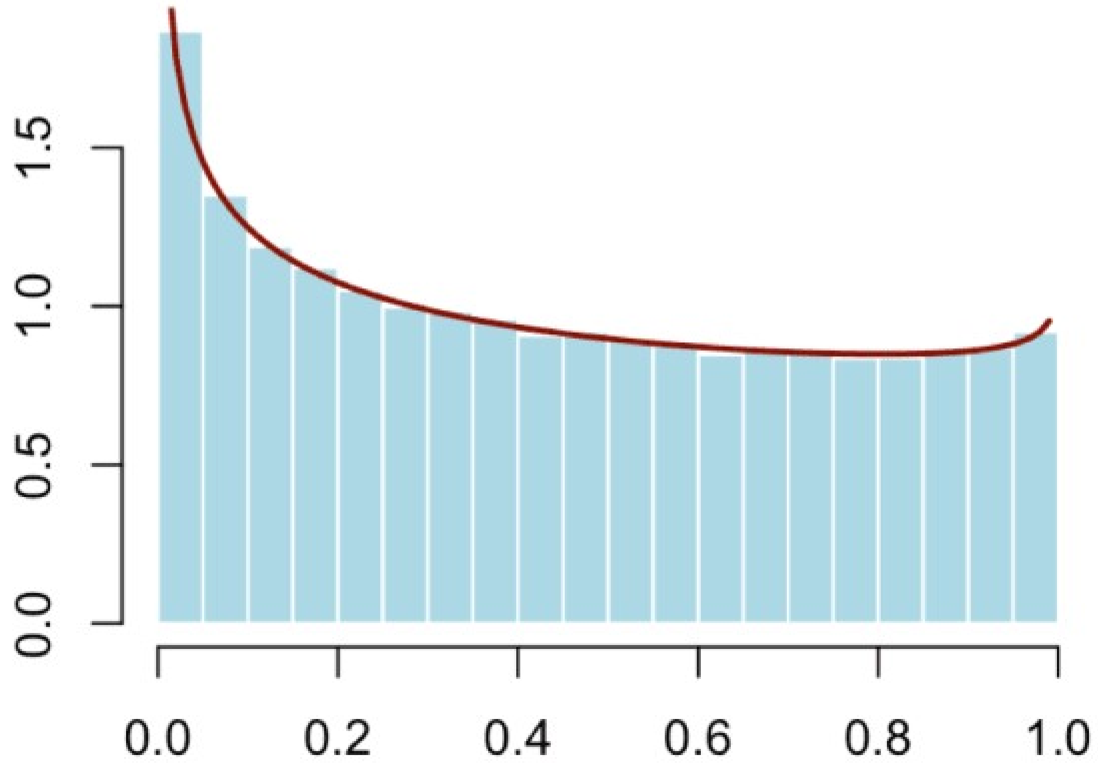
> When a measure becomes a target, it ceases to be a good measure

i.e. we cannot assess goodness of fit using the same data as the ones used to estimate parameters. So here, why not use some hold-out (or cross-validation) procedure : split the dataset in two parts, \(\{x_1,\cdots,x_k\}\) (with \(kKolmogorov–Smirnov statistics on it to test if [latex]x_i\)'s are drawn from some Gaussian distribution. More precisely, will the \(p\)-value computed using the standard Kolmogorov–Smirnov procedure be ok here. Here, I tried two scenarios, \(k/n\) being either \(1/3\) or \(2/3\),

```
p = matrix(NA,1e5,4)
for(s in 1:1e5){
X = rnorm(n,0,1)
p[s,1] = ks.test(X,"pnorm",0,1)$p.value
p[s,2] = ks.test(X,"pnorm",mean(X),sd(X))$p.value
p[s,3] = ks.test(X[1:200],"pnorm",mean(X[201:300]),sd(X[201:300]))$p.
value
p[s,4] = ks.test(X[201:300],"pnorm",mean(X[1:200]),sd(X[1:200]))$
p.value
}
```

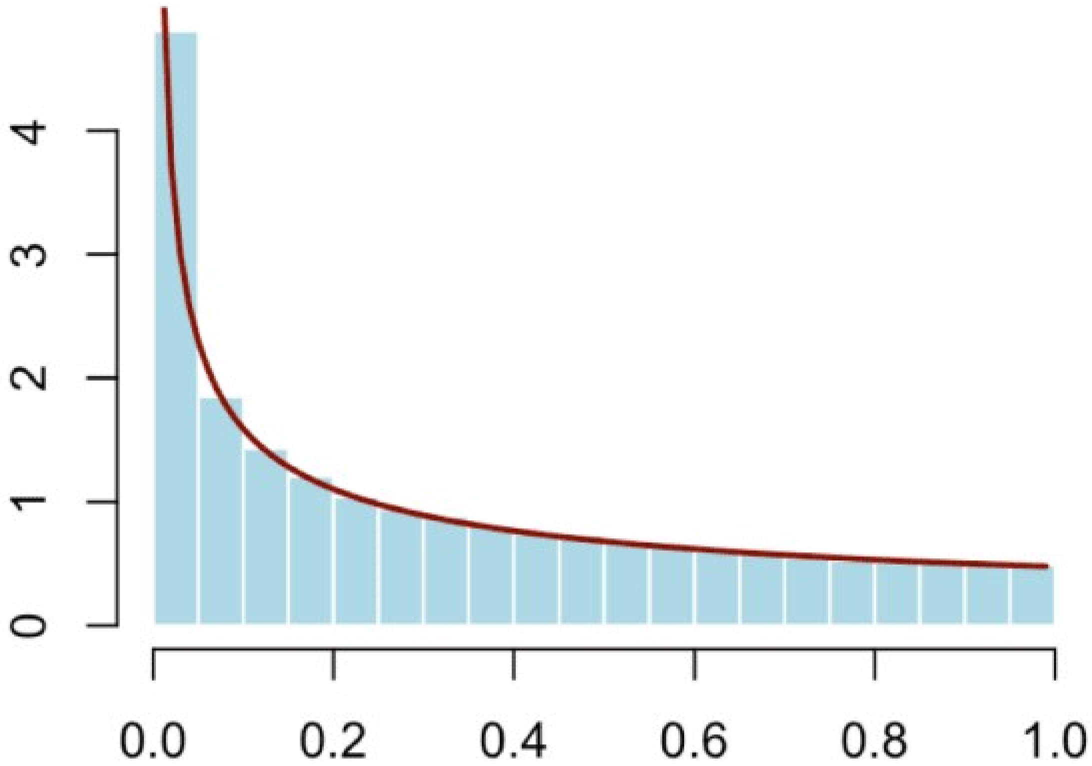
Again, we can visualize the distributions of \(p\)-values, in the case where \(1/3\) of the data is used to estimate \(\mu\) and \(\sigma^2\), and \(2/3\) of the data is used to test

```
fit.dist = fitdist(p[,3],"beta")
hist(p[,3],probability = TRUE,main="",xlab="",ylab="")
vu=seq(0,1,by=.01)
vv=dbeta(vu,shape1 = fit.dist$estimate[1], shape2 =
fit.dist$estimate[2])
lines(vu,vv,col="dark red", lwd=2)
```




and in the case where \(2/3\) of the data is used to estimate \(\mu\) and \(\sigma^2\), and \(1/3\) of the data is used to test

```
fit.dist = fitdist(p[,4],"beta")
hist(p[,4],probability = TRUE,main="",xlab="",ylab="")
vu=seq(0,1,by=.01)
vv=dbeta(vu,shape1 = fit.dist$estimate[1], shape2 =
fit.dist$estimate[2])
lines(vu,vv,col="dark red", lwd=2)
```




Observe here that we (wrongly) reject too frequently $H_0$, since the $p$-values are below 5% in 25% of the scenarios, in the first case (less data used to estimate), and 9% of the scenarios, in the second case (less data used to test)

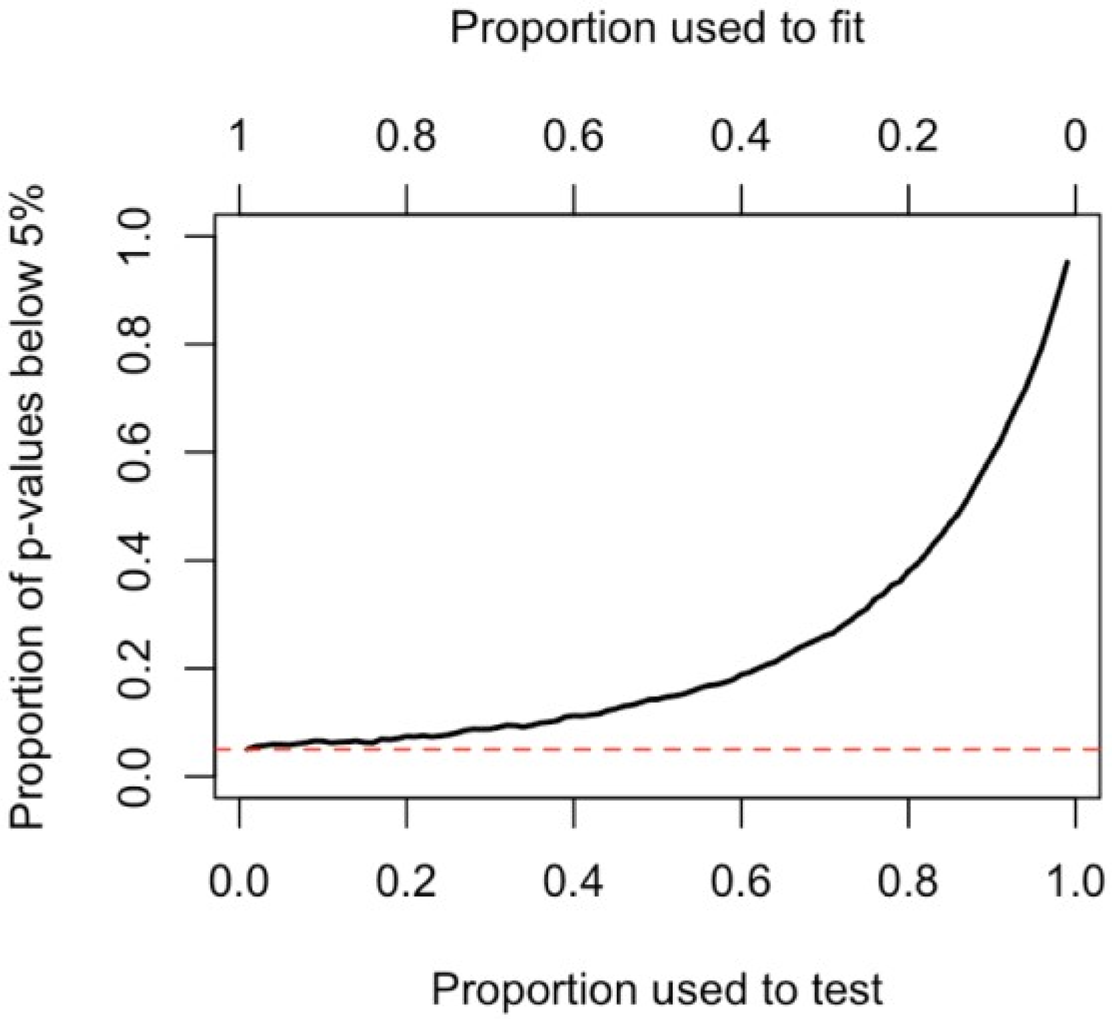
```
mean(p[,3]<=.05)
[1] 0.24168
mean(p[,4]<=.05)
[1] 0.09334
```

We can actually compute that probability as a function of $k/n$

```
n=300
p = matrix(NA,1e4,99)
for(s in 1:1e4){
  X = rnorm(n,0,1)
  KS = function(p) ks.test(X[1:(p*n)],"pnorm",
mean(X[(p*n+1):n]),sd(X[(p*n+1):n]))$p.value
  p[s,] = Vectorize(KS)((1:99)/100)
}
```

The evolution of the probability is the following

```
prob5pc = apply(p,2,function(x) mean(x<=.05))
plot((1:99)/100,prob5pc)
```




so, it looks like we can use some sort of hold-out procedure to test for $H_0:X_i \sim\mathcal{N(\mu,\sigma^2)}$, for some $\mu$ and $\sigma^2$, using Kolmogorov–Smirnov test with $\mu=\hat\mu$ and $\sigma^2=\hat\sigma^2$ but the proportion of data used to estimate those quantities should be (much) larger that the one used to compute the statistics. Otherwise, we clearly reject too frequently $H_0$.