# MATH 559: Bayesian Theory and Methods

## Selection with the Poisson model

Suppose that a model is to be constructed under an assumption of exchangeability with the following components:

- Data $y_1, \ldots, y_n$ recorded;
- $f_Y(y;\theta) \equiv Poisson(\theta)$.
- $\pi_0(\theta_0) \equiv Gamma(\alpha_n, \beta_n) \equiv Gamma(\alpha_0, \beta_0)$ with $\alpha_0 = 2$, $\beta_0 = 0.1$.
- True data generating model: $Poisson(\theta_0)$ with $\theta_0 = 5$.

Then we have

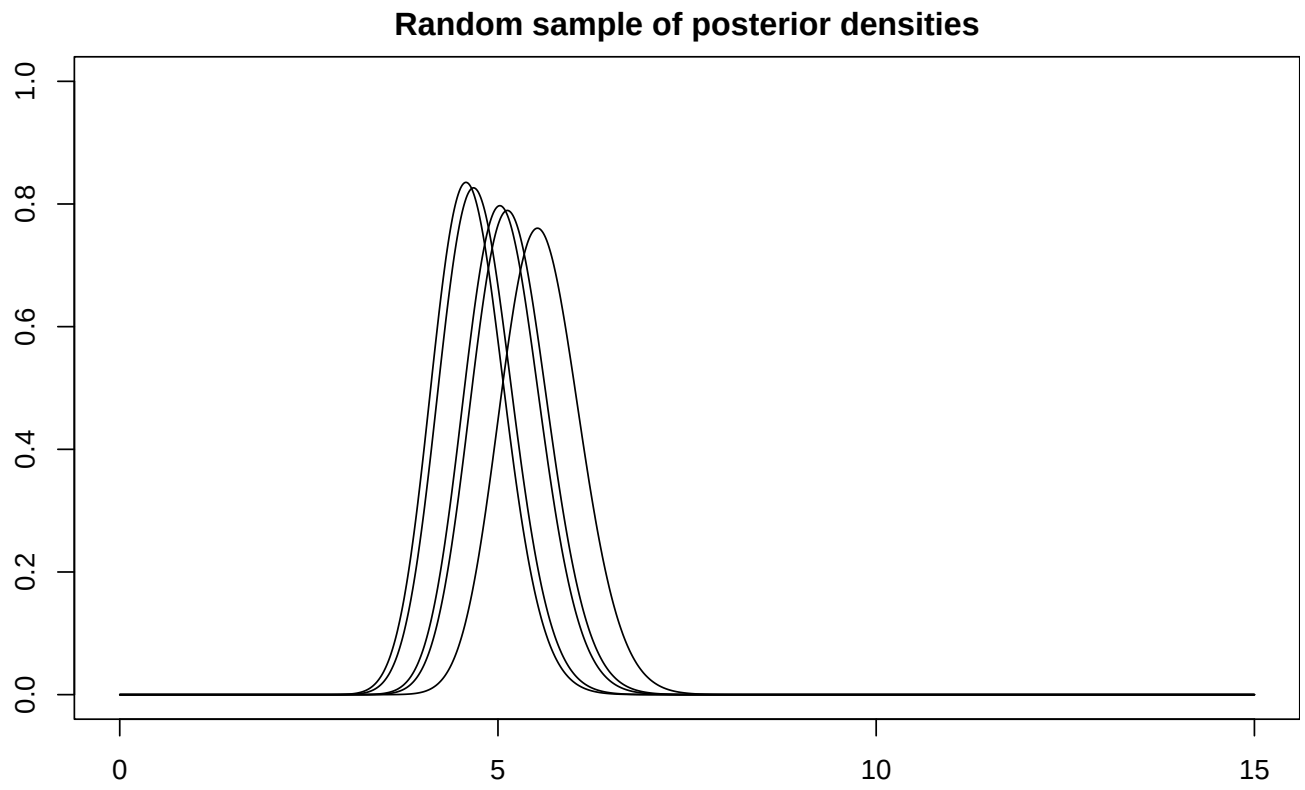
- Posterior: $\pi_n(\theta) \equiv Gamma(\alpha_0 + s_n, \beta_0 + n)$ where $s_n = \sum_{i=1}^{n} y_i$. For the random variable version,
$$\widetilde{\pi}_n(\theta) \equiv Gamma(\alpha_0 + S_n, \beta_0 + n)$$
and recall from elementary results that $S_n \sim Poisson(n\theta_0)$.
- Prior predictive:
$$f_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n) = \int_0^\infty \prod_{i=1}^{n} f_Y(y_i;\theta)\pi_0(\theta)\, d\theta = \int_0^\infty \frac{1}{\prod_{i=1}^{n} y_i!}\theta^{s_n}e^{-n\theta}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\theta^{\alpha_0-1}e^{-\beta_0\theta}\, d\theta$$
$$= \frac{1}{\prod_{i=1}^{n} y_i!}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\int_0^\infty \theta^{s_n+\alpha_0-1}e^{-(n+\beta_0)\theta}\, d\theta$$
$$= \frac{1}{\prod_{i=1}^{n} y_i!}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\frac{\Gamma(\alpha_0+s_n)}{(n+\beta_0)^{\alpha_0+s_n}}$$
- Posterior predictive: changing $(\alpha_0, \beta_0)$ to $(\alpha_n, \beta_n)$ in the above formula, we have for a single $y = 0, 1, 2, \ldots$
$$p_n(y) = \int_0^\infty f_Y(y;\theta)\pi_n(\theta)\, d\theta = \frac{1}{y!}\frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)}\frac{\Gamma(\alpha_n+y)}{(\beta_n+1)^{\alpha_n+y}} = \frac{\Gamma(\alpha_n+y)}{y!\Gamma(\alpha_n)}\left(\frac{\beta_n}{\beta_n+1}\right)^{\alpha_n}\left(\frac{1}{\beta_n+1}\right)^{y}$$
which is a form of the Negative Binomial distribution.

The KL divergence between $f^*(y)$ and $p_n(y)$ is

$$KL(f^*, p_n) = \sum_{y=0}^{\infty}\log\left(\frac{f^*(y)}{p_n(y)}\right)f^*(y) = \sum_{y=0}^{\infty}\log(f^*(y))f^*(y) - \sum_{y=0}^{\infty}\log(p_n(y))f^*(y). \qquad (\diamondsuit)$$

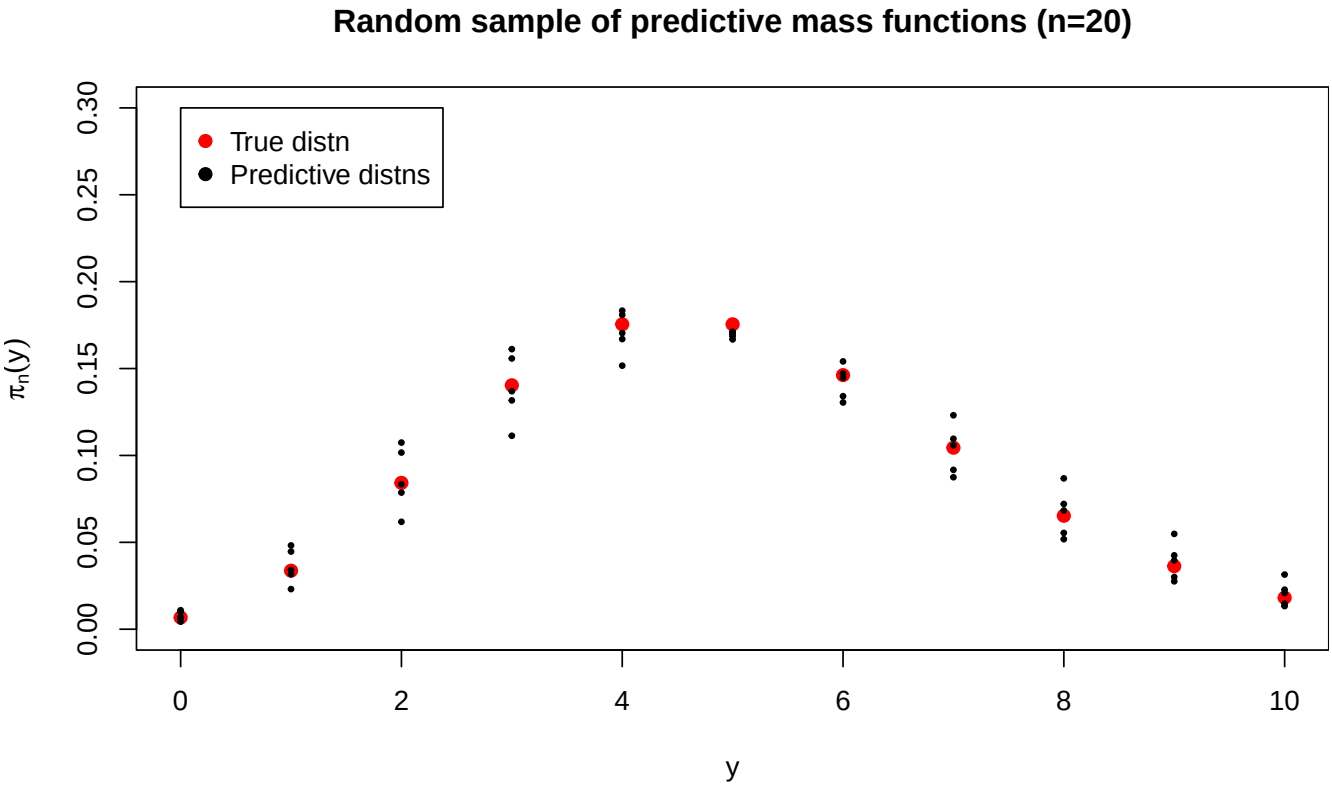
The first term in ($\diamondsuit$) is a constant which does not depend on the inference model. The random variable version $KL(f^*, \widetilde{p}_n)$ can also be considered.

```
set.seed(2134)
n<-20;nreps<-5
th0<-5
a0<-2; b0<-0.1
bn<-b0+n
par(mar=c(3,3,2,0))
xv<-seq(0,15,by=0.01)
yv<-dnorm(xv,0,1)
plot(xv,yv,type='n',main='Random sample of posterior densities',ylim=range(0,1))
for(irep in 1:nreps){
    sn<-sum(rpois(n,th0))
    an<-a0+sn
    yv<-dgamma(xv,an,bn)
    lines(xv,yv)
}
```

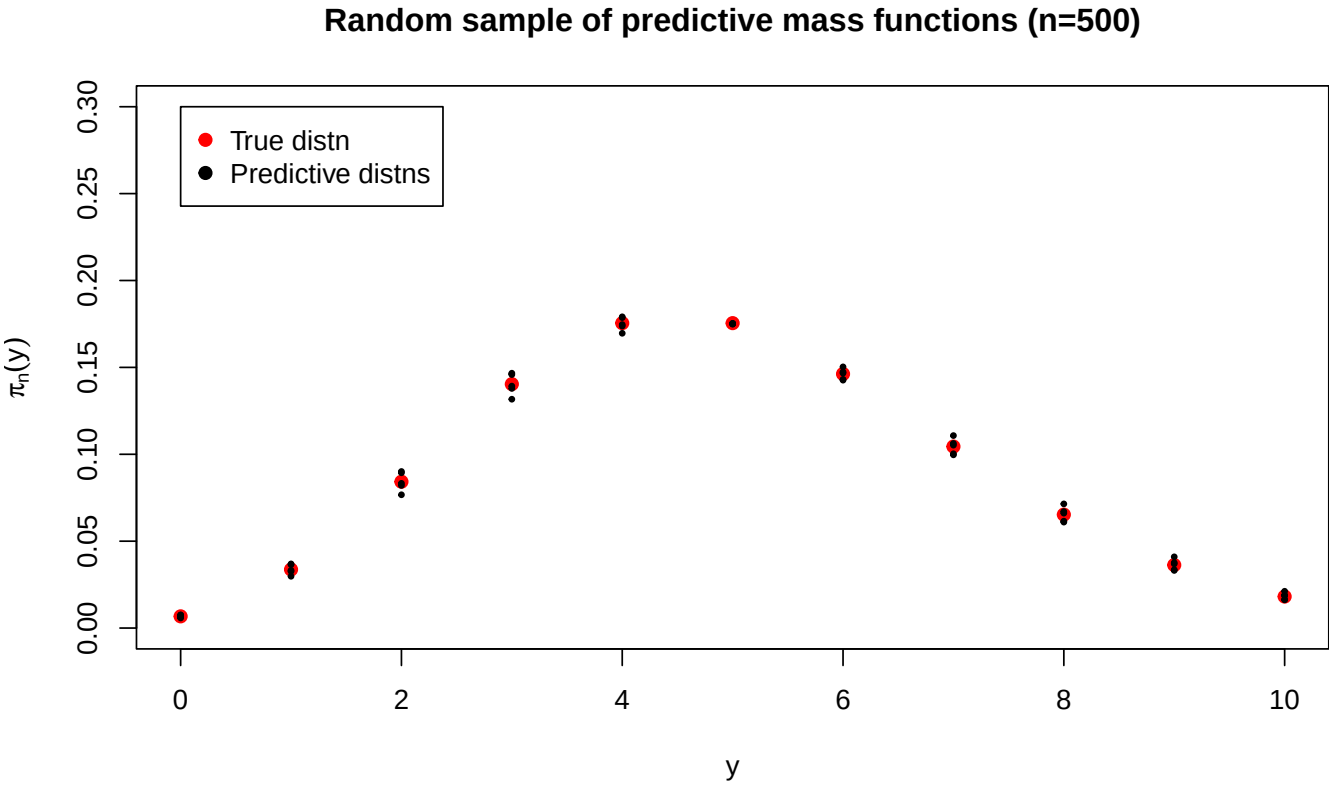



The posterior predictive distribution $p_n(y)$ can be regarded as a Bayesian estimate of the true data generating mass function $f^*(y) \equiv Poisson(5)$.

```
xv<-seq(0,10,by=01)
yv<-dpois(xv,th0)
par(mar=c(4,4,4,0))
plot(xv,yv,type='p',main='Random sample of predictive mass functions (n=20)',
    ylim=range(0,0.3),pch=19,col='red',xlab='y',ylab=expression(pi[n](y)))
set.seed(2134)
for(irep in 1:nreps){
    sn<-sum(rpois(n,th0))
    an<-a0+sn
    pv<-exp(lgamma(an+xv)-lgamma(xv+1)-lgamma(an)+an*log(bn/(1+bn))-xv*log(1+bn))
    points(xv,pv,pch=19,cex=0.4)
}
legend(0,0.3,c('True distn','Predictive distns'),col=c('red','black'),pch=19)
```




For $n = 500$, we practically recover $f^*(y)$ in each replicate.



The following statistics can be used for model selection: in this case, the quantities can often be computed analytically, and can sometimes be computed numerically. Recall that in all cases, the $Y_1, \ldots, Y_n$ are deemed to have been sampled from $f^* \equiv Poisson(\theta_0)$ with $\theta_0 = 5$.

- **Training loss:** The *training loss*, $T_n$, is a measure that approximates the KL divergence based on the sample

$$T_n \equiv T(Y_1, \ldots, Y_n) = -\frac{1}{n} \sum_{i=1}^{n} \log \widetilde{p}_n(Y_i)$$

  which can be regarded as a sample-based estimator of the second term in ($\diamondsuit$).

- **Generalization loss:** The *generalization loss*, $G_n$, is the second term in ($\diamondsuit$):

$$G_n \equiv G(Y_1, \ldots, Y_n) = -\sum_{y=0}^{\infty} \log \widetilde{p}_n(y) f^*(y).$$

- **Entropy:** The first term in ($\diamondsuit$) is often denoted $-S$, where

$$S = -\sum_{y=0}^{\infty} \log(f^*(y)) f^*(y)$$

  and is termed the *entropy* of $f^*$. With $f^*(y) \equiv Poisson(\theta_0)$,

$$S = -\sum_{y=0}^{\infty} (-\theta_0 + y \log \theta_0 - \log y!) \frac{e^{-\theta_0} \theta_0^y}{y!}$$

  which we can compute by truncating the sum:

```
y<-c(0:10000)
Sval<--sum((-th0+y*log(th0)-lgamma(y+1))*dpois(y,th0))
Sval
## [1] 2.204395
y<-c(0:100000)
Sval<--sum((-th0+y*log(th0)-lgamma(y+1))*dpois(y,th0))
Sval
## [1] 2.204395
```

  The quantity $G_n - S$ is termed the *generalization error*.

- **Cross-validation loss:** The *cross-validation* loss, $C_n$, is defined by

$$C_n = -\frac{1}{n} \sum_{i=1}^{n} \log \widetilde{p}_n^{(-i)}(Y_i).$$

  In this case, the leave-one-out posterior predictive takes the form

$$p_n^{(-i)}(y) = \frac{\Gamma(\alpha_0 + s_n - y_i + y)}{y! \Gamma(\alpha_0 + s_n - y_i)} \left( \frac{\beta_0 + n - 1}{\beta_0 + n} \right)^{\alpha_0 + s_n - y_i} \left( \frac{1}{\beta_0 + n} \right)^y$$

  We have the alternative formula

$$C_n = \frac{1}{n} \sum_{i=1}^{n} \log \mathbb{E}_{\widetilde{\pi}_n} \left[ \frac{1}{f_Y(Y_i; \theta)} \right].$$

  Here, for $y = 0, 1, 2, \ldots$

$$\begin{aligned} \mathbb{E}_{\widetilde{\pi}_n} \left[ \frac{1}{f_Y(y; \theta)} \right] &= \int_0^{\infty} \frac{y!}{e^{-\theta} \theta^y} \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \theta^{\alpha_n - 1} e^{-\beta_n \theta} \, d\theta \\ &= \frac{y! \beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \int_0^{\infty} \theta^{(\alpha_n - y) - 1} e^{-(\beta_n - 1)\theta} \, d\theta \\ &= \frac{y! \beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \frac{\Gamma(\alpha_n - y)}{(\beta_n - 1)^{\alpha_n - y}} \end{aligned}$$

- **WAIC:** The *widely applicable information criterion* (or WAIC), $W_n$, is defined by

$$W_n = T_n + \frac{1}{n}\sum_{i=1}^{n} \text{Var}_{\widetilde{\pi}_n}[\log f_Y(Y_i;\theta)]$$

  where $T_n$ is the training loss. We must compute the variance using Monte Carlo.

- **Marginal likelihood (or prior predictive):**

$$z_n = \frac{1}{\prod_{i=1}^{n} y_i!}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\frac{\Gamma(\alpha_0+s_n)}{(n+\beta_0)^{\alpha_0+s_n}}$$

  with the equivalent random variable version, $Z_n$, and with $F_n = -\log Z_n$ is the *free energy*. We can also report the standardized version

$$\overline{F}_n = -\frac{1}{n}\log Z_n.$$

In large samples, the quantities $T_n$, $G_n$, $C_n$ and $W_n$ are numerically very similar, and have the same limiting value.

**Simulation Study:**

```
set.seed(2134)
n<-20;nreps<-1000
bni<-n-1+b0;

posterior.predictive<-function(yv,nv,a0v,b0v){

    snv<-sum(yv)
    anv<-a0v+snv
    bnv<-b0v+nv
    pv<-exp(lgamma(anv+yv)-lgamma(yv+1)-lgamma(anv)+anv*log(bnv/(1+bnv))-yv*log(1+bnv))
    return(pv)
}

exp.posterior.predictive<-function(yv,nv,a0v,b0v,th0v){

    snv<-sum(yv)
    anv<-a0v+snv
    bnv<-b0v+nv
    y<-c(0:1000)
    lpv<-lgamma(anv+y)-lgamma(y+1)-lgamma(anv)+anv*log(bnv/(1+bnv))-y*log(1+bnv)

    return(-sum(lpv*dpois(y,th0v)))
}

posterior.predictive.i<-function(yv,nv,a0v,b0v){

    snv<-sum(yv)
    anv<-a0v+snv-yv
    bnv<-b0v+nv-1
    pv<-exp(lgamma(anv+yv)-lgamma(yv+1)-lgamma(anv)+anv*log(bnv/(1+bnv))-yv*log(1+bnv))
    return(pv)
}

Y<-matrix(rpois(n*nreps,th0),ncol=n)
Snv<-rowSums(Y)

Tn<--log(apply(Y, 1, posterior.predictive, nv=n, a0v=a0, b0v=b0))
Tn<-apply(Tn,2,mean)
```

```r
Gn<-apply(Y, 1, exp.posterior.predictive,nv=n,a0v=a0,b0v=b0,th0v=th0)

Cn<--log(apply(Y, 1, posterior.predictive.i, nv=n, a0v=a0, b0v=b0))
Cn<-apply(Cn,2,mean)

posterior.reciprocal<-function(yv,nv,a0v,b0v){
    snv<-sum(yv)
    anv<-a0v+snv
    bnv<-b0v+nv
    lpv<-lgamma(yv+1)+anv*log(bnv)-lgamma(anv)+lgamma(anv-yv)-(anv-yv)*log(bnv-1)
    return(lpv)
}

Cn2<-apply(Y,1,posterior.reciprocal,nv=n,a0v=a0,b0v=b0)
Cn2<-apply(Cn2,2,mean)

variance.term<-function(yv,nv,a0v,b0v,N=10000){

    #Monte Carlo calculation
    snv<-sum(yv)
    anv<-a0v+snv
    bnv<-b0v+nv

    thv<-rgamma(N,anv,bnv)
    d<-outer(yv,thv,dpois, log=T)

    return(mean(apply(d,1,var)))
}
Wn<-Tn+apply(Y,1,variance.term, nv=n,a0v=a0,b0v=b0)

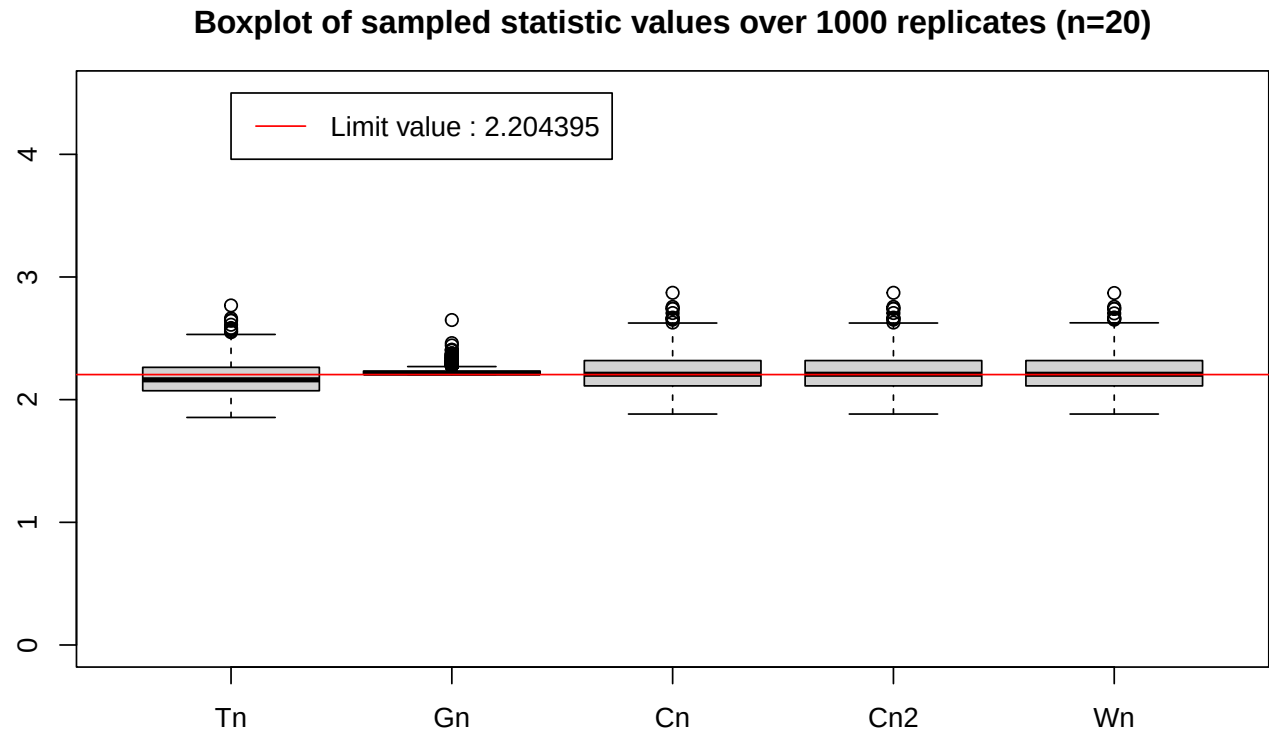
prior.predictive<-function(yv,nv,a0v,b0v){

    snv<-sum(yv)
    anv<-a0v+snv
    bnv<-b0v+nv

    pv<-exp(-sum(lgamma(yv+1))+a0v*log(b0v)-lgamma(a0v)+lgamma(anv)-anv*log(bnv))
    return(pv)
}

Zn<-apply(Y,1,prior.predictive, nv=n,a0v=a0,b0v=b0)
Fn<--log(Zn)
c(mean(Zn),mean(Fn))

+ [1]  0.00000000000000000007631806 46.744190732683925659785018 07
```




Means across the 1000 replicate data sets for $n = 20$: recall that the limiting value is $S = 2.204395$

```
+       Tn       Gn       Cn      Cn2       Wn
+ 2.175754 2.228619 2.223507 2.223507 2.223350
```

Larger sample size: $n = 200$

```
set.seed(2134)
n<-200;nreps<-1000
bni<-n-1+b0;


Y<-matrix(rpois(n*nreps,th0),ncol=n)
Snv<-rowSums(Y)

Tn<--log(apply(Y, 1, posterior.predictive, nv=n, a0v=a0, b0v=b0))
Tn<-apply(Tn,2,mean)

Gn<-apply(Y, 1, exp.posterior.predictive,nv=n,a0v=a0,b0v=b0,th0v=th0)

Cn<--log(apply(Y, 1, posterior.predictive.i, nv=n, a0v=a0, b0v=b0))
Cn<-apply(Cn,2,mean)

Cn2<-apply(Y,1,posterior.reciprocal,nv=n,a0v=a0,b0v=b0)
Cn2<-apply(Cn2,2,mean)

Wn<-Tn+apply(Y,1,variance.term, nv=n,a0v=a0,b0v=b0)

Zn<-apply(Y,1,prior.predictive, nv=n,a0v=a0,b0v=b0)
Fn<--log(Zn)
c(mean(Zn),mean(Fn))
```

```
+ [1]   0.000000000
```