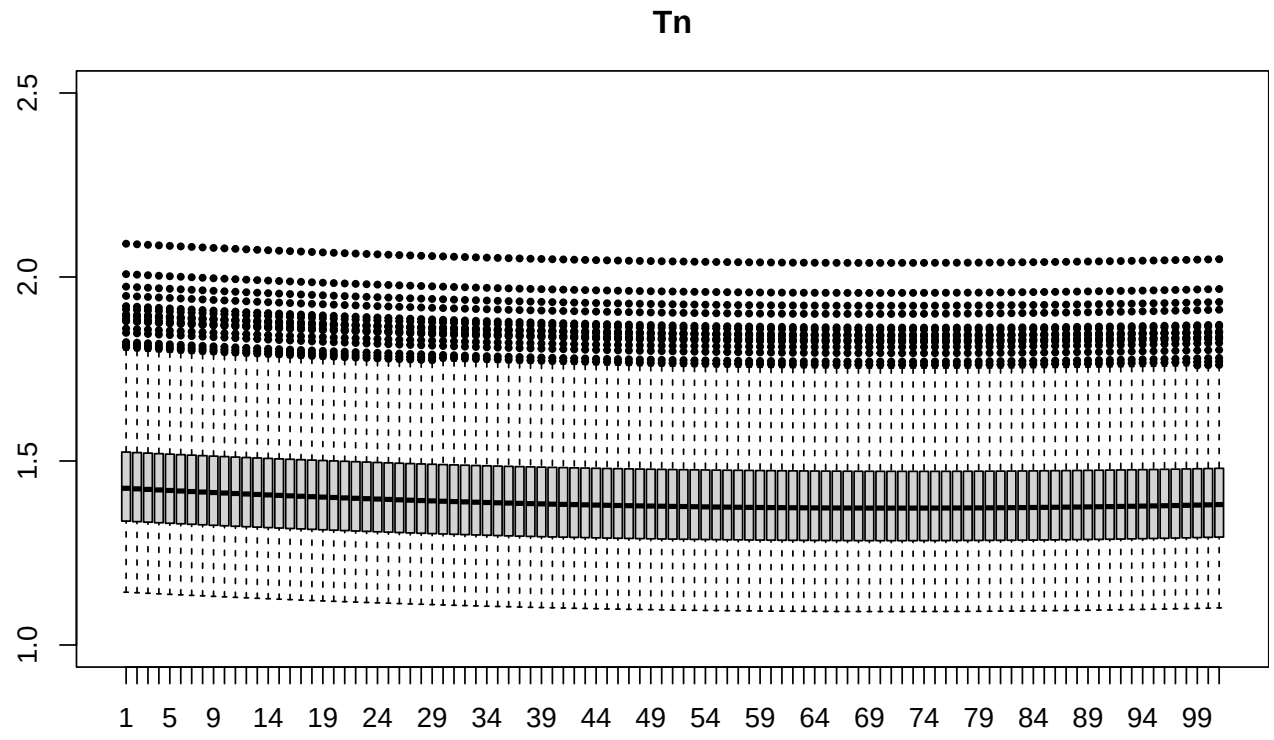
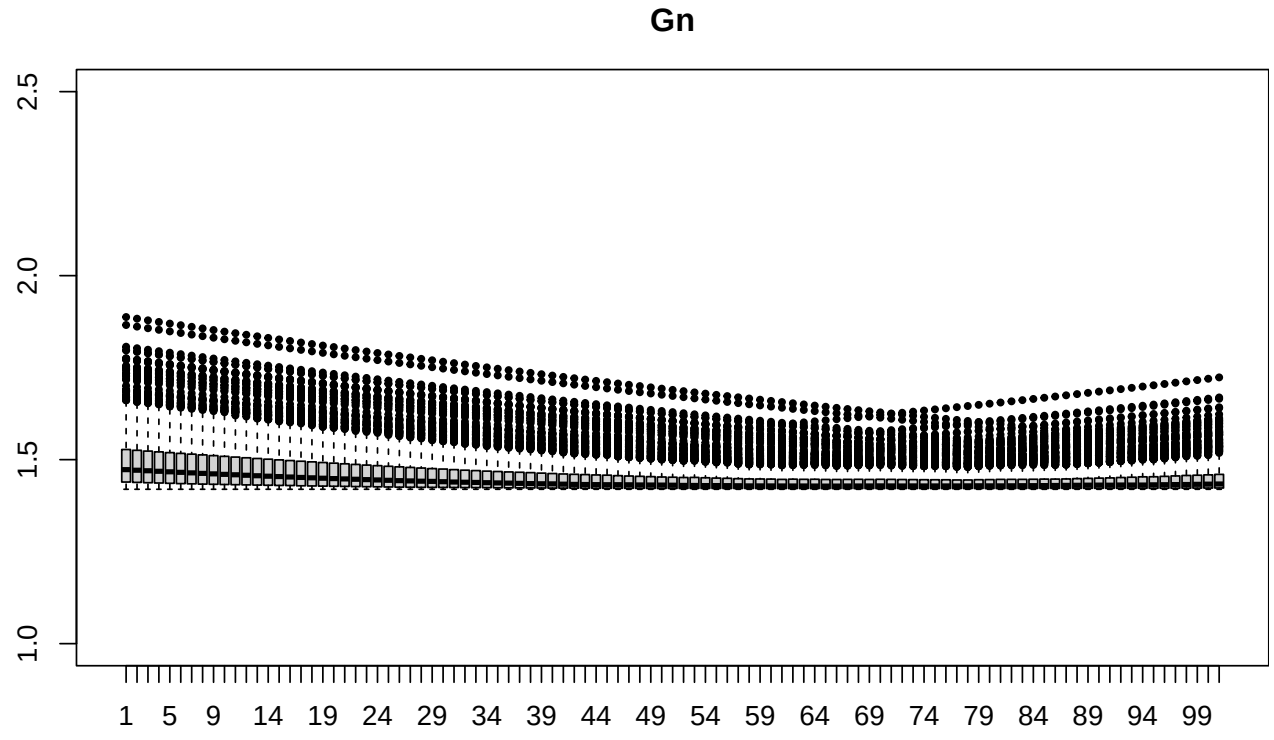
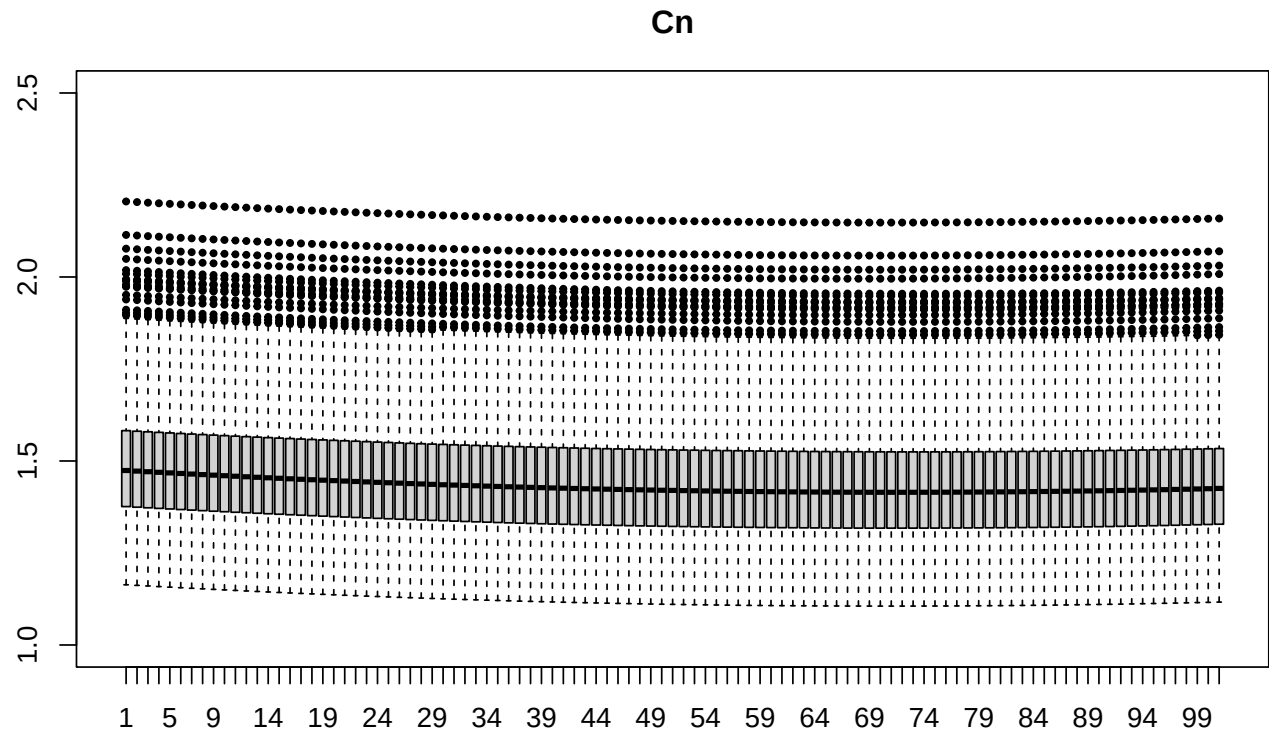
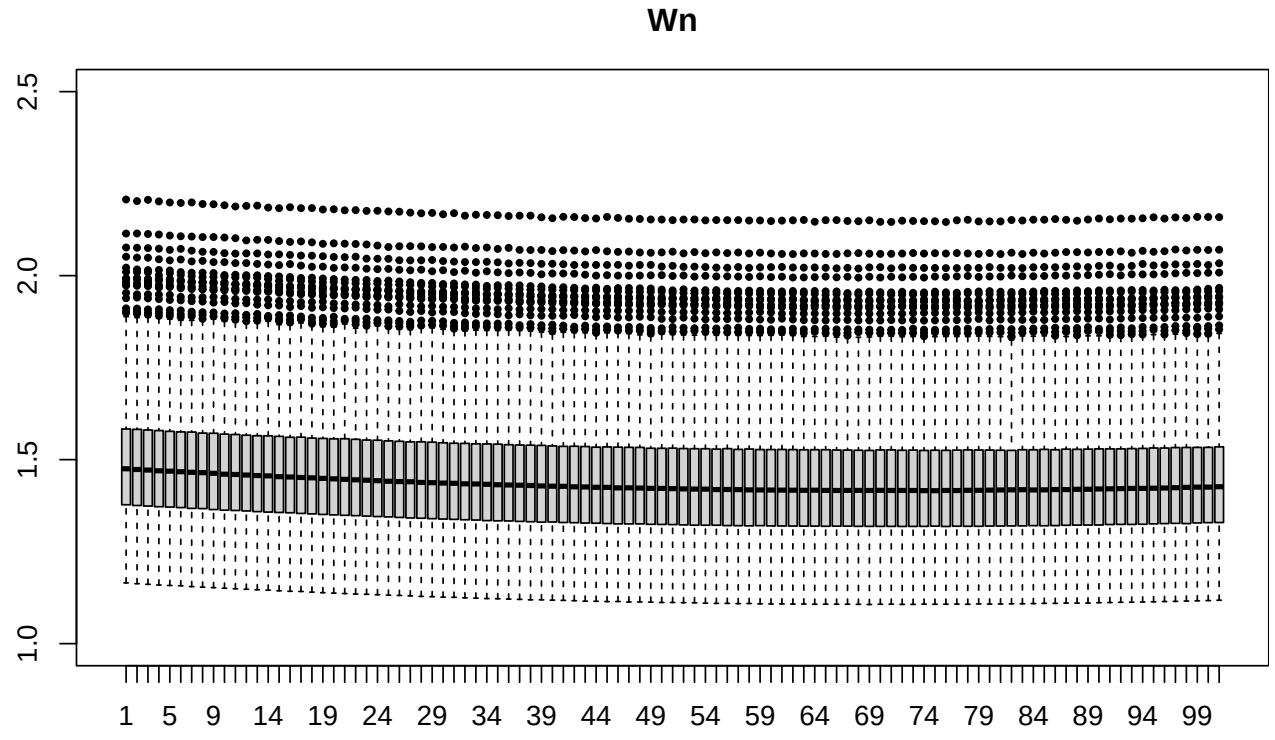
MATH 559 : BAYESIAN THEORY AND METHODS
SELECTION WITH THE NORMAL MODEL: EXPLORING DIFFERENT PRIORS

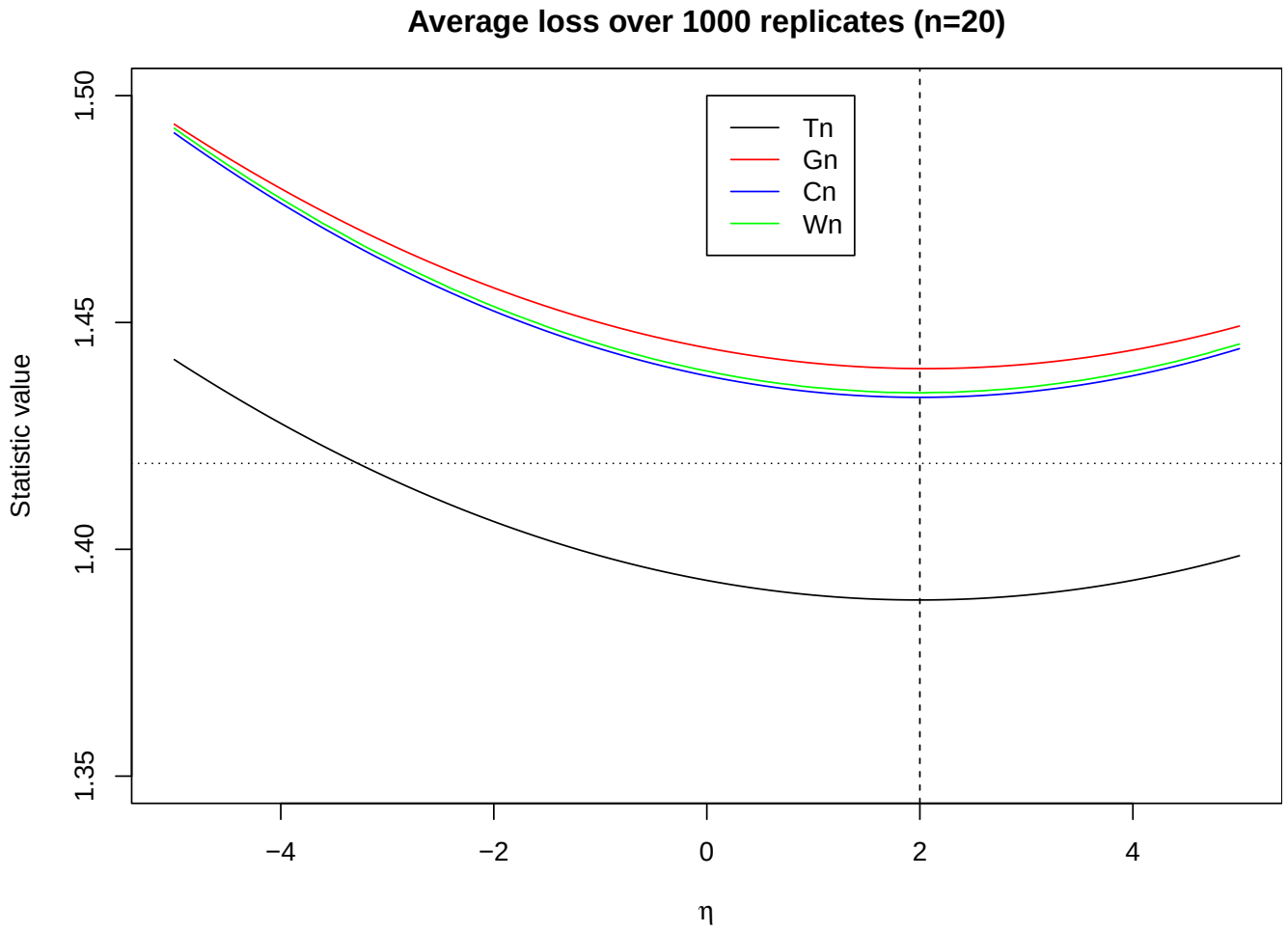
```
set.seed(2134)
n<-20;nreps<-1000
mu0<-2;sigma0<-1
eta<-0; lambda<-1
dsq<-function(xv,ev,lv){
    dv<-xv*0
    for(j in 1:length(xv)){
        dv[j]<-xv[j]-(sum(xv[-j])+ev*lv)/(length(xv)-1+lv)
    }
    return(sum(dv^2))
}
ssq<-function(xv){
    return(sum((xv-mean(xv)^2)))
}
variance.term<-function(xv,ev,lv,N=10000){

    #Monte Carlo calculation
    en<-(sum(xv)+ev*lv)/(length(xv)+lv)
    ln<-length(xv)+lv
    mu<-rnorm(N,en,sqrt(1/ln))
    d<-outer(xv,mu,'-')

    return(mean(apply(dnorm(d,log=T),1,var)))
}
Y<-matrix(rnorm(n*nreps,2,1),ncol=n)

etavec<-seq(-5,5,by=0.1)
lambda.n<-n+lambda; lambda.n1<-lambda.n/(1+lambda.n)
lambda.ni<-n-1+lambda; lambda.ni1<-lambda.ni/(1+lambda.ni)
const<-0.5*log(2*pi)-0.5*log(lambda.n1)
stat.mat<-array(0,c(ncol=length(etavec),nreps,4))
print(dim(stat.mat))

+ ncol
+  101 1000    4

for(il in 1:length(etavec)){
    eta<-etavec[il]
    eta.n<-(n*apply(Y,1,mean)+eta*lambda)/lambda.n
    Tn<-const+0.5*lambda.n1*apply((Y-eta.n)^2,1,sum)/n
    Gn<-const+0.5*lambda.n1*(1+(eta.n-mu0)^2)
    Cn<-const+0.5*lambda.ni1*apply(Y,1,dsq,ev=eta,lv=lambda)/n
    Wn<-Tn+apply(Y,1,variance.term,ev=eta,lv=lambda)
    stat.mat[il,,]<-cbind(Tn,Gn,Cn,Wn)
}
```


Tn
2.5
2.0
1.5
1.0
1 5 9 14 19 24 29 34 39 44 49 54 59 64 69 74 79 84 89 94 99


Gn
2.5
2.0
1.5
1.0
1 5 9 14 19 24 29 34 39 44 49 54 59 64 69 74 79 84 89 94 99

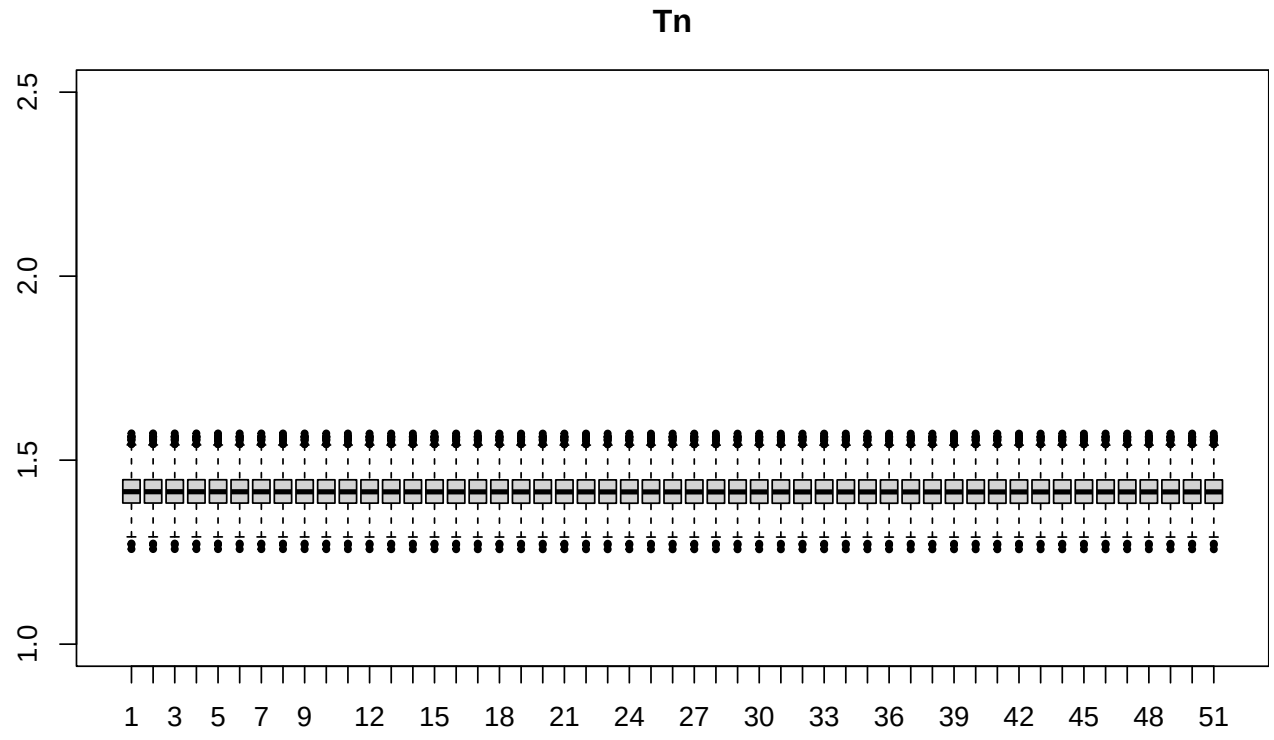

Cn
2.5
2.0
1.5
1.0
1 5 9 14 19 24 29 34 39 44 49 54 59 64 69 74 79 84 89 94 99

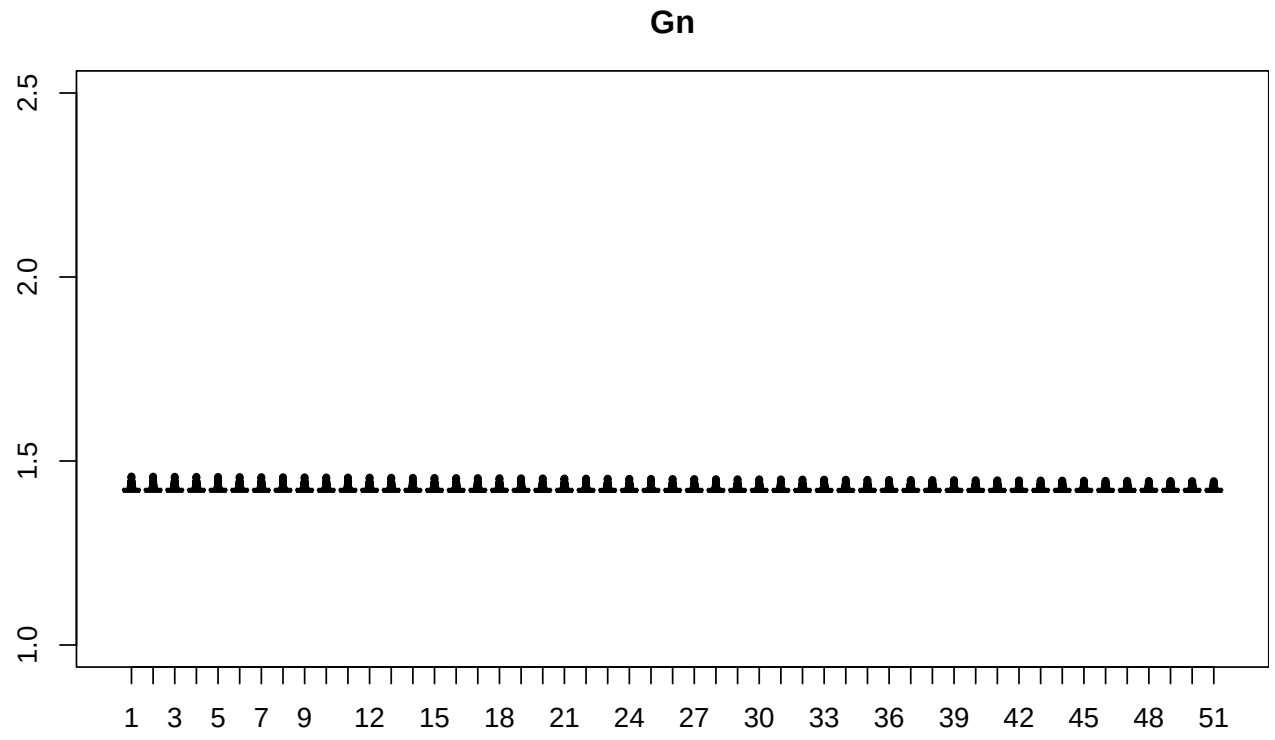

Wn
2.5
2.0
1.5
1.0
1 5 9 14 19 24 29 34 39 44 49 54 59 64 69 74 79 84 89 94 99

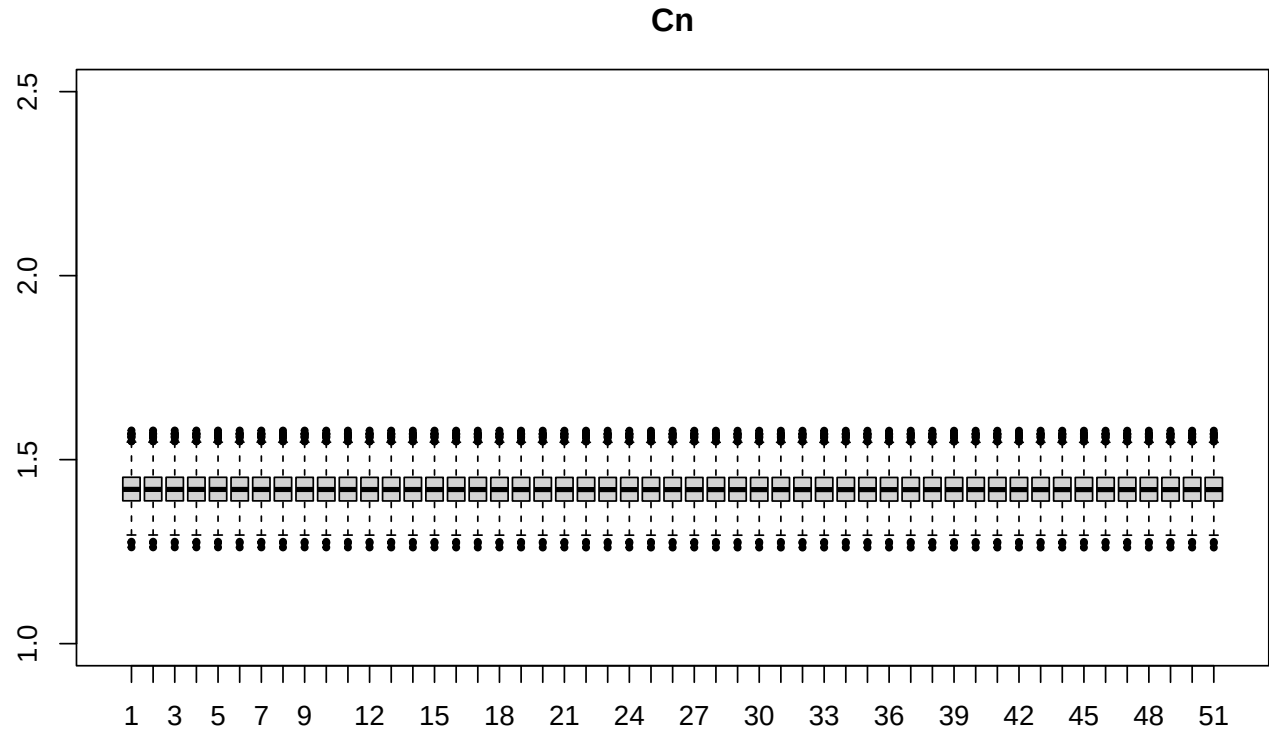


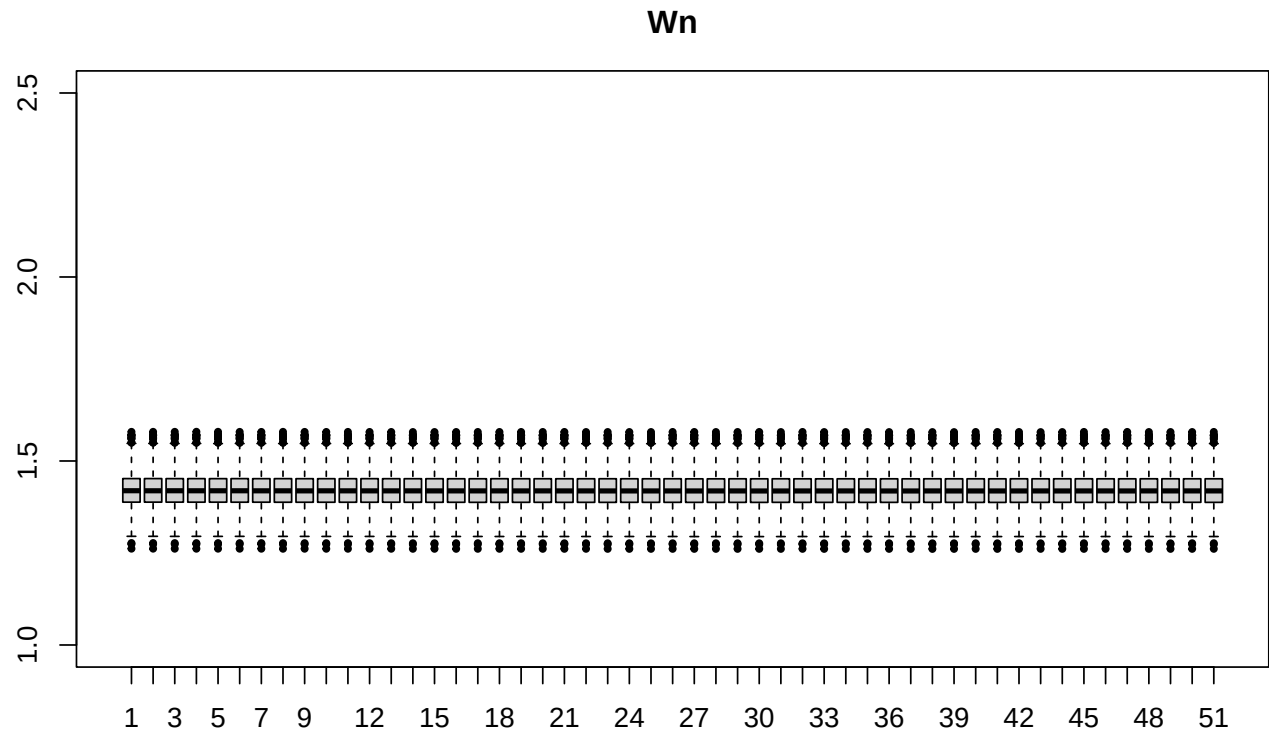

```r
set.seed(2134)
n<-200;nreps<-1000
mu0<-2;sigma0<-1
eta<-0; lambda<-1
dsq<-function(xv,ev,lv){
    dv<-xv*0
    nx<-length(xv)
    tx<-sum(xv)
    for(j in 1:nx){
        dv[j]<-xv[j]-((tx-xv[j])+ev*lv)/(nx-1+lv)
    }
    return(sum(dv^2))
}
ssq<-function(xv){
    return(sum((xv-mean(xv)^2)))
}
variance.term<-function(xv,ev,lv,N=10000){

    #Monte Carlo calculation
    en<-(sum(xv)+ev*lv)/(length(xv)+lv)
    ln<-length(xv)+lv
    mu<-rnorm(N,en,sqrt(1/ln))
    d<-outer(xv,mu,'-')

    return(mean(apply(dnorm(d,log=T),1,var)))
}
```

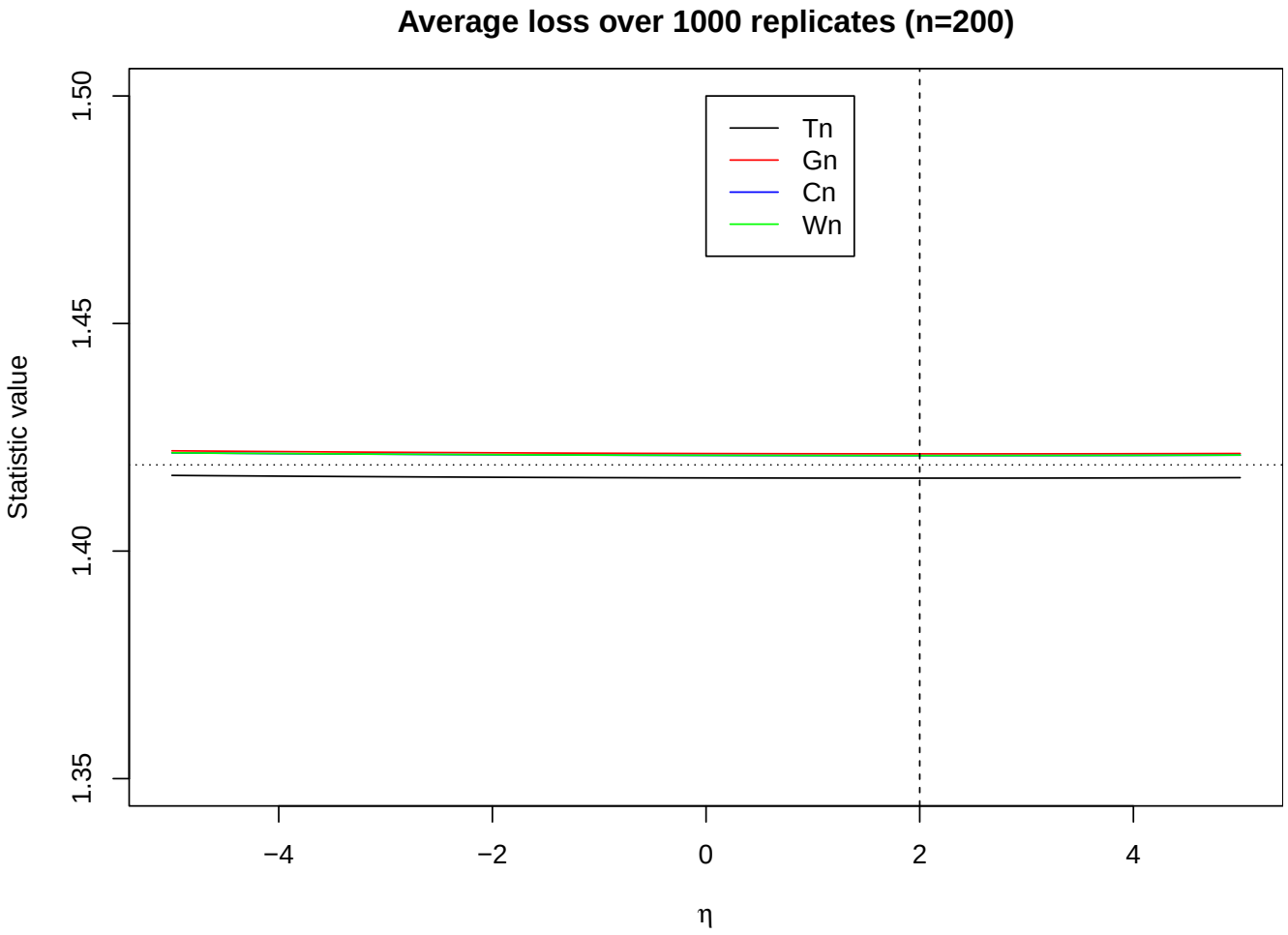
```r
Y<-matrix(rnorm(n*nreps,2,1),ncol=n)

etavec<-seq(-5,5,by=0.2)
lambda.n<-n+lambda; lambda.n1<-lambda.n/(1+lambda.n)
lambda.ni<-n-1+lambda; lambda.ni1<-lambda.ni/(1+lambda.ni)
const<-0.5*log(2*pi)-0.5*log(lambda.n1)
stat.mat<-array(0,c(ncol=length(etavec),nreps,4))
print(dim(stat.mat))

+ ncol
+   51 1000    4

for(il in 1:length(etavec)){
    eta<-etavec[il]
    eta.n<-(rowSums(Y)+eta*lambda)/lambda.n
    Tn<-const+0.5*lambda.n1*rowMeans((Y-eta.n)^2)
    Gn<-const+0.5*lambda.n1*(1+(eta.n-mu0)^2)
    Cn<-const+0.5*lambda.ni1*apply(Y,1,dsq,ev=eta,lv=lambda)/n
    Wn<-Tn+apply(Y,1,variance.term,ev=eta,lv=lambda)
    stat.mat[il,,]<-cbind(Tn,Gn,Cn,Wn)
}
```




Gn
2.5
2.0
1.5
1.0
1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51


Cn
2.5
2.0
1.5
1.0
1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51


Wn
2.5
2.0
1.5
1.0
1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51


Average loss over 1000 replicates (n=200)
Statistic value
1.50
1.45
1.40
1.35
−4
−2
0
2
4
η
Tn
Gn
Cn
Wn