

simulations

2025-03-10

Générer les données

Générer les fonctions

```
gener_tte=function(mu, lambda=1){  
  X=rexp(n, rate=lambda)  
  C=rexp(n, rate = mu)  
  TT=pmin(X,C)  
  delta=as.numeric(X==TT)  
  return(list(TT, delta))  
}  
  
gener_continue=function(mean=1, sd=0){  
  X=abs(rnorm(n, mean=mean, sd=sd))  
  return(X)  
}  
  
gener_binom=function(prob){  
  X=rbinom(n, size = 1, prob = prob)  
  return(X)  
}
```

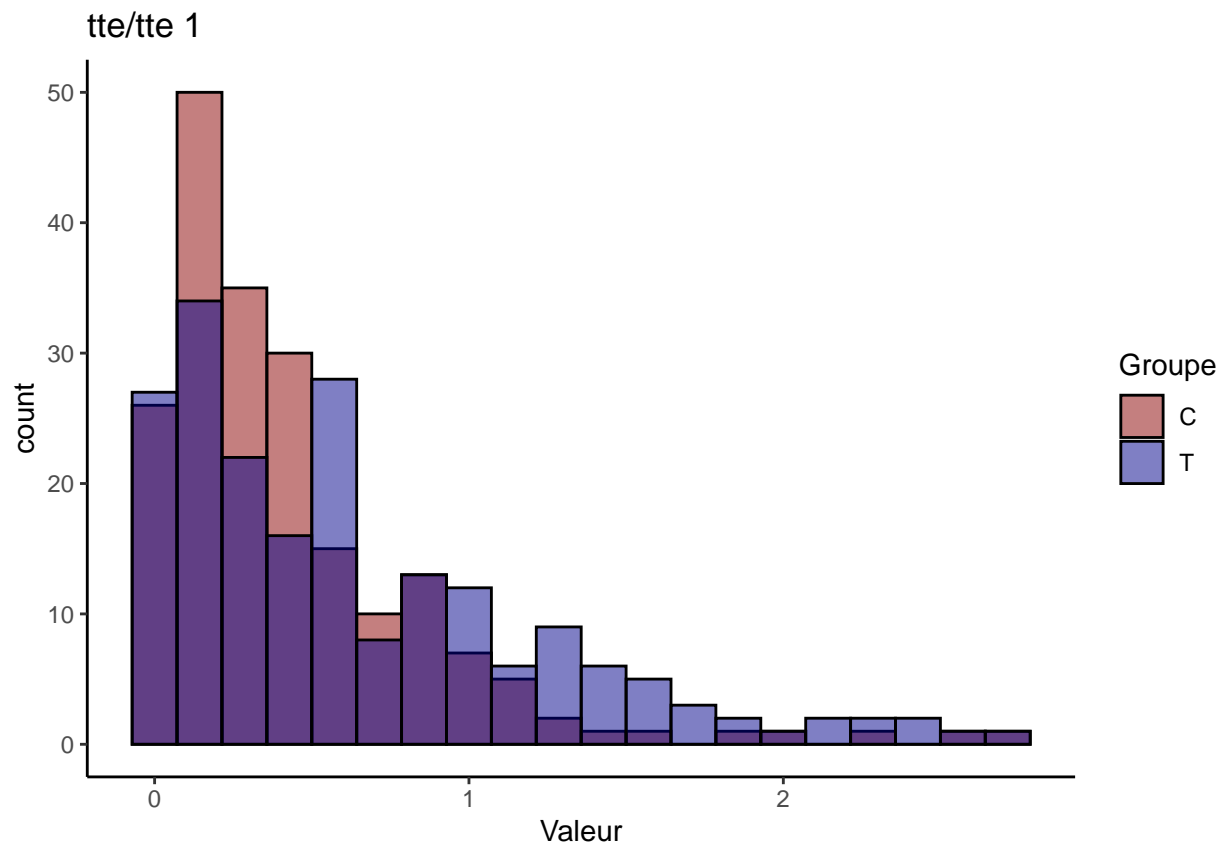
Différents scénario

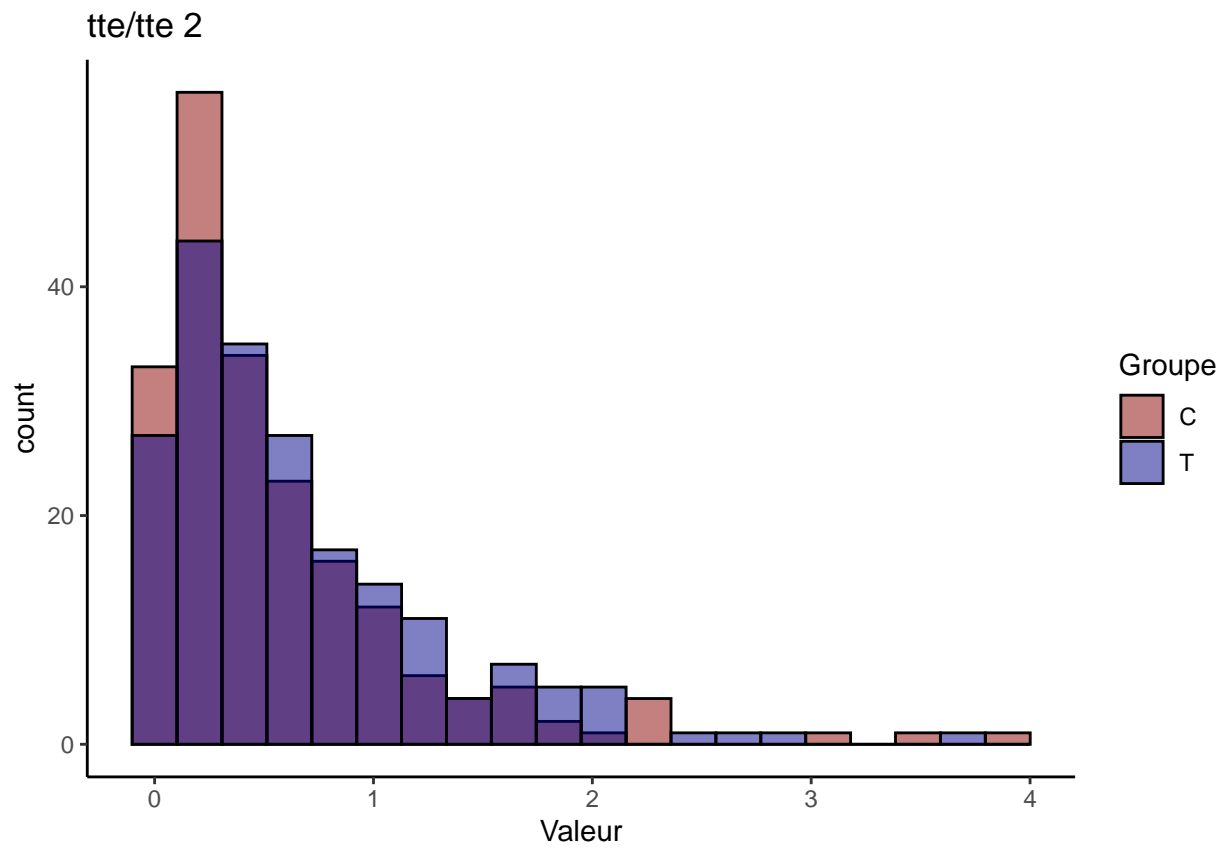
Je prend comme généralités que : - pour les binomiales, 1 correspond à l'évènement d'intérêt et qu'il est meilleur d'avoir 0 que 1 \ - pour les tte, plus un évènement est censuré moins il est bon (à voir encore) \ - pour les continues, il n'y a que des valeurs positives et une valeur plus grande est préférable \

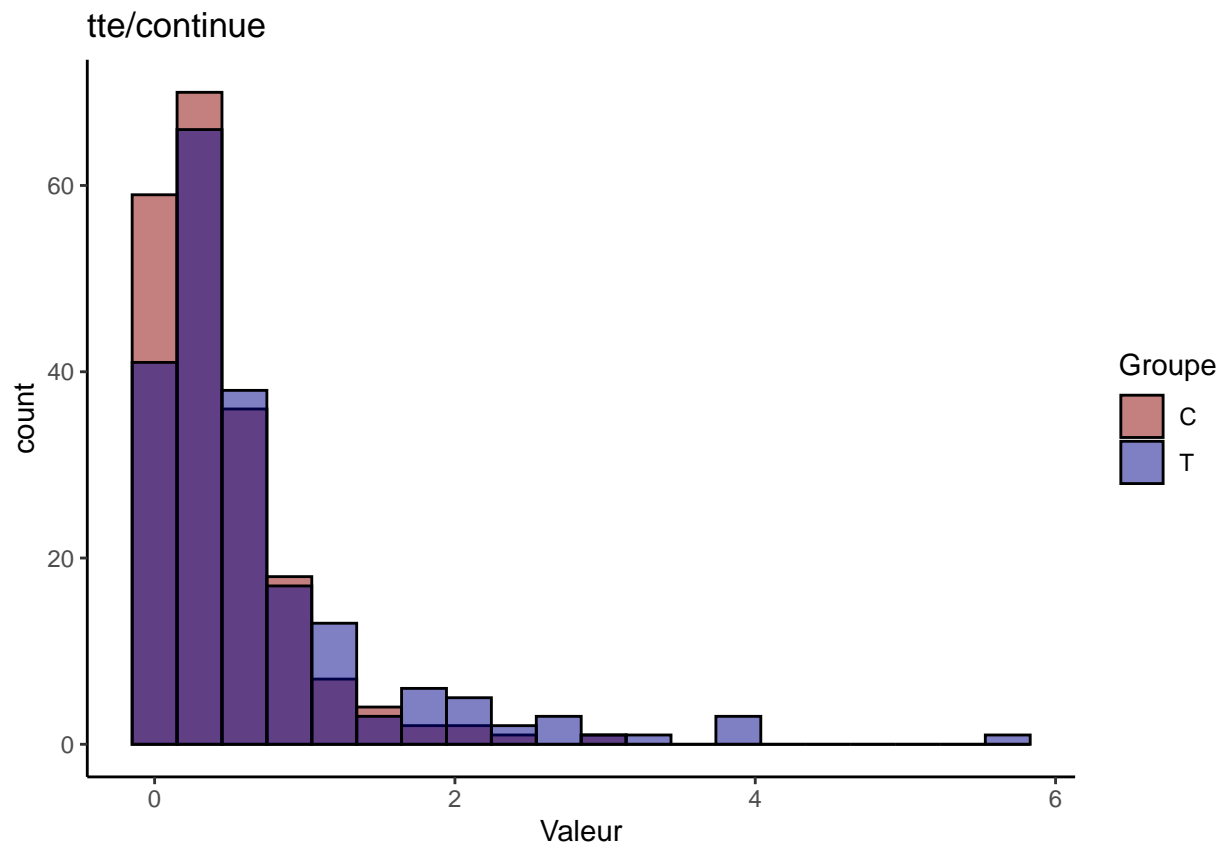
Les notations des vecteurs T et C sont les suivantes : T_scénario_cas de figure (resp C_scénario_cas de figure) où le scénario varie entre 1 et 3 et les cas de figure entre 1 et 6 comme suit : - 1 : tte/tte \ - 2 : tte/continue \ - 3 : tte/binaire \ - 4 : continue/continue \ - 5 : continue/binaire \ - 6 : binaire/binaire \

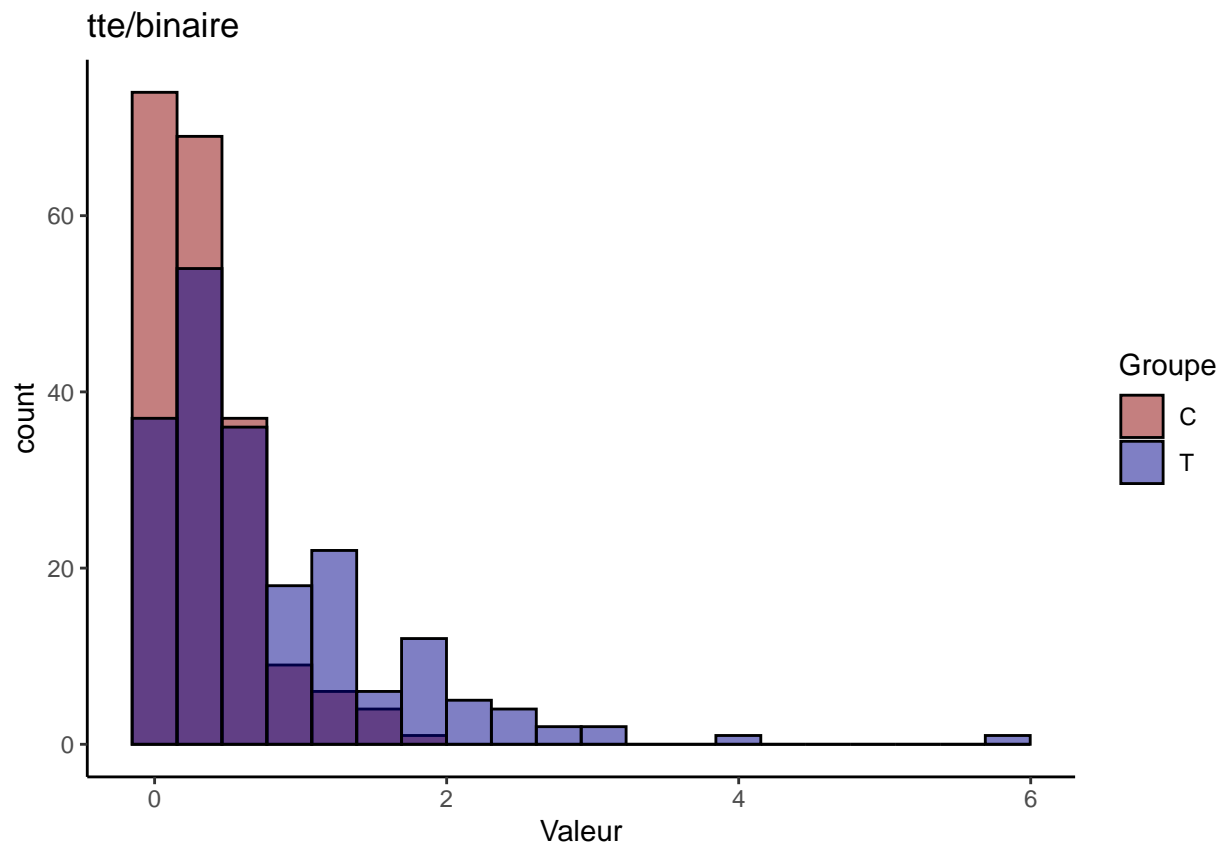
Scénario 1 : T uniformément meilleur que C

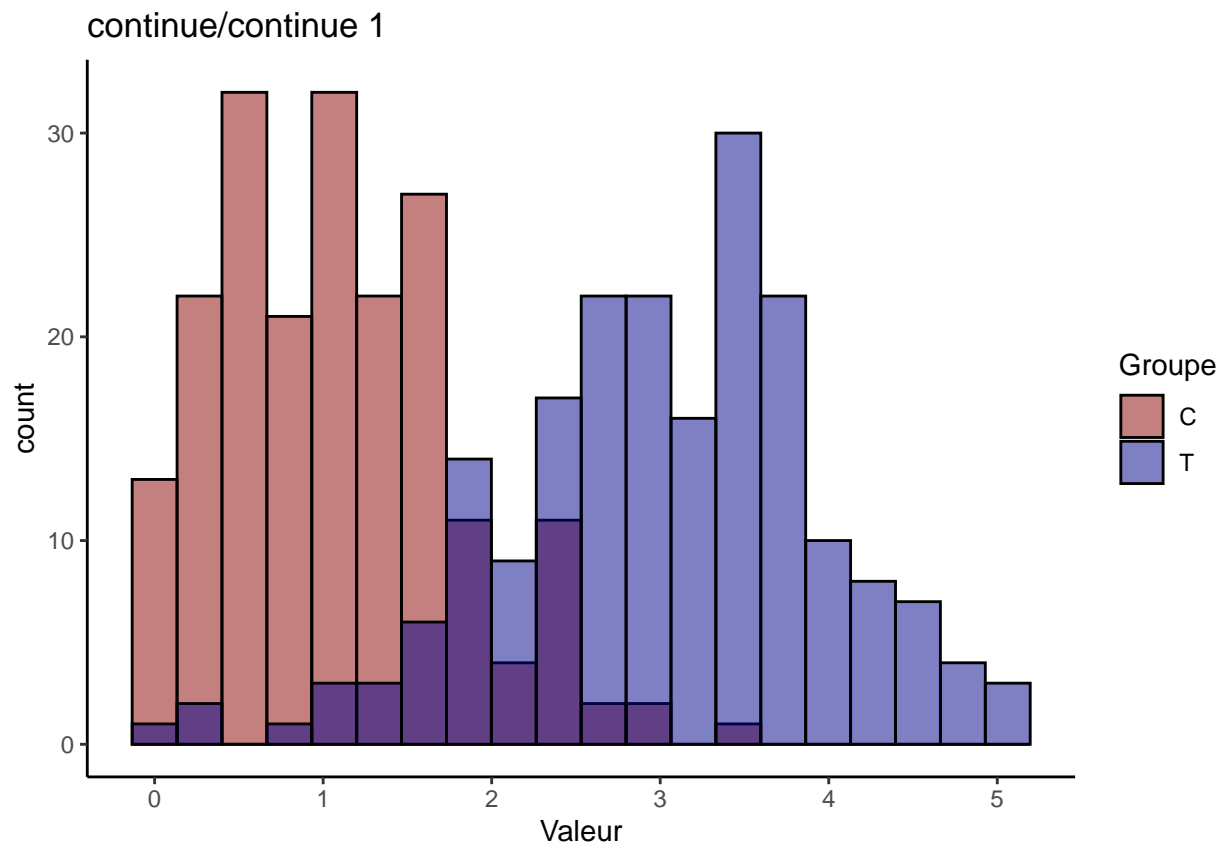
La visualisation sert à vérifier si les valeurs correspondent bien au scénario choisis

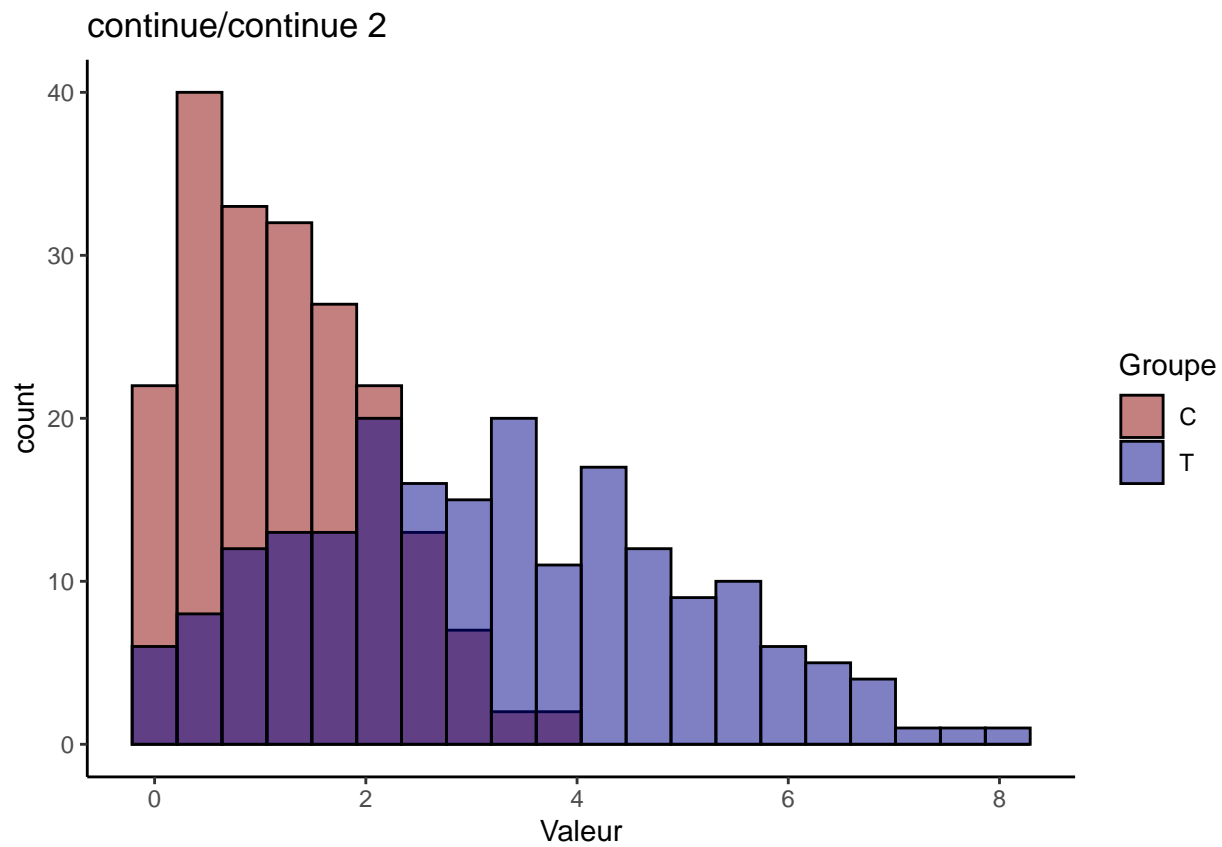


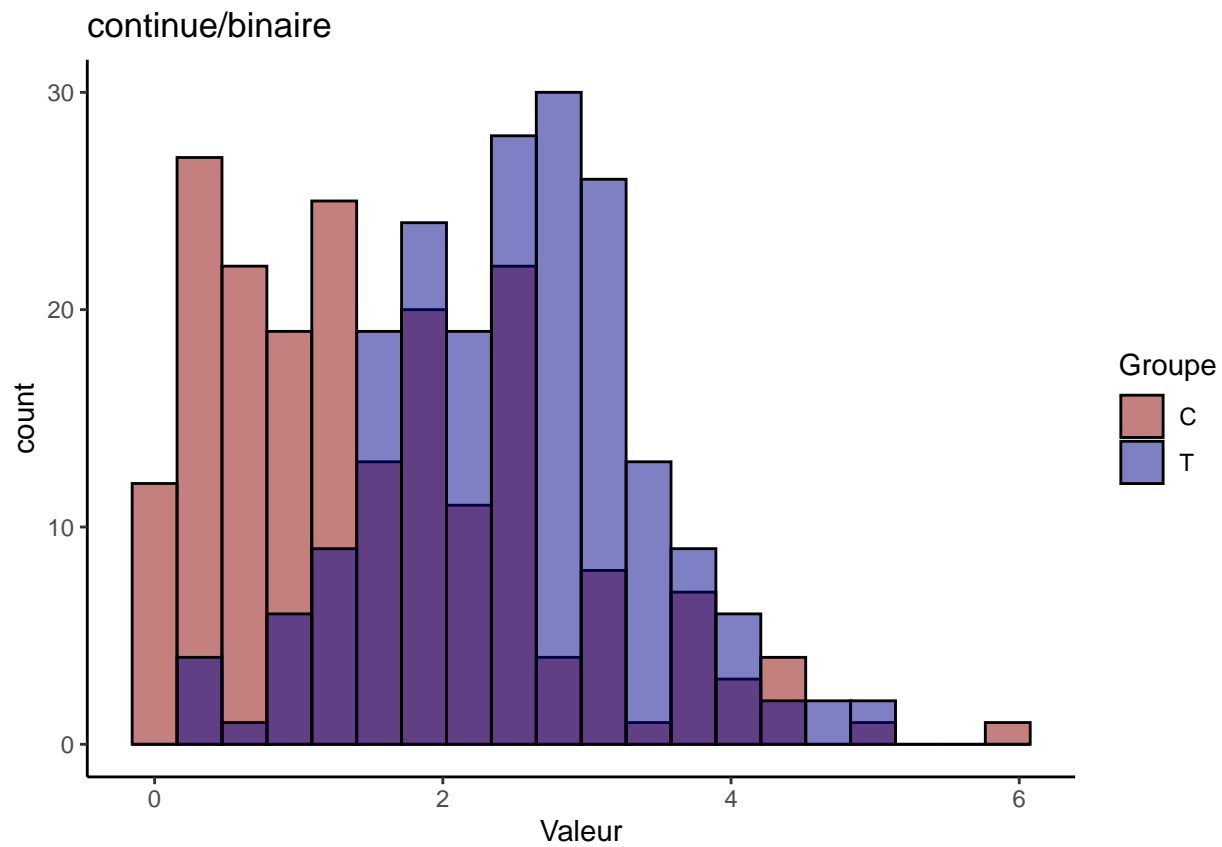


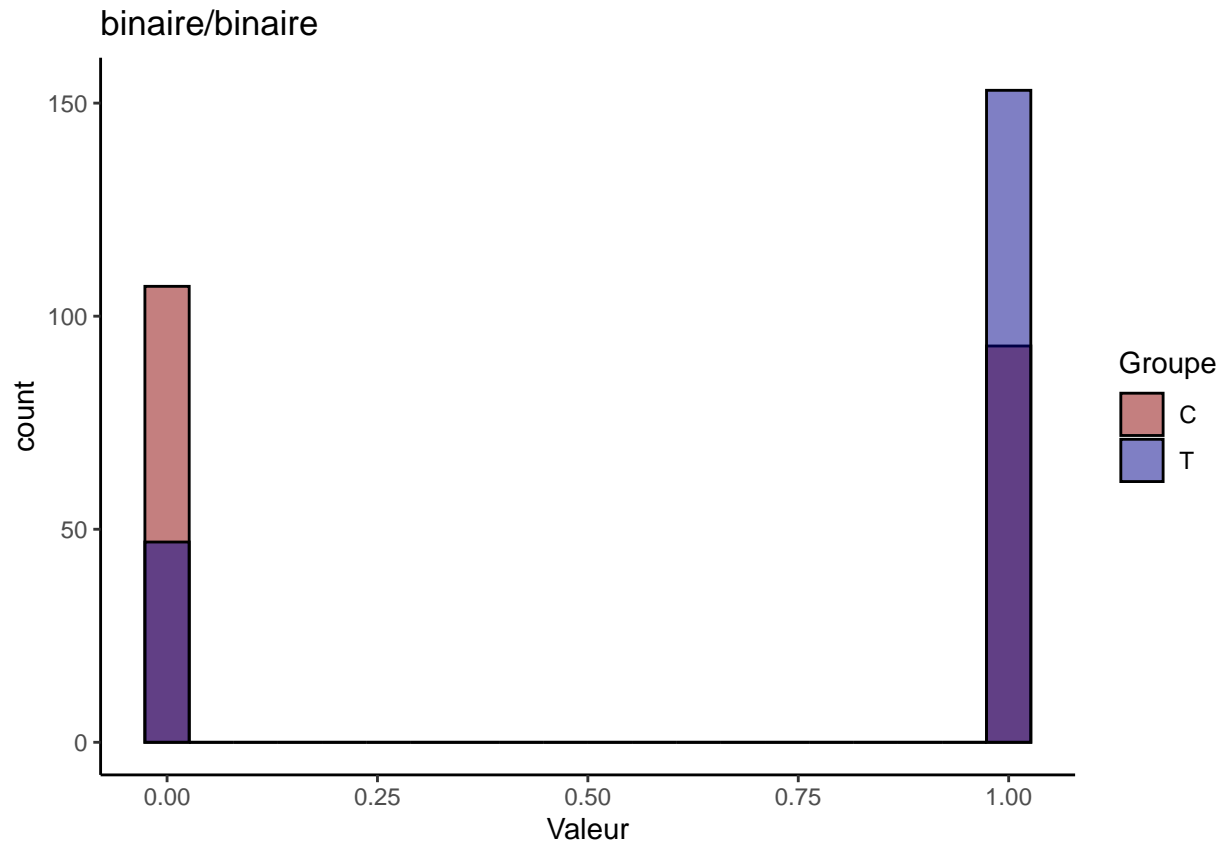












Scénario 2 : C uniformément meilleur que T

Scénario 3 : T similaire à C

GPC

```

affect_value_GPC=function(vecteur1,vecteur2,w=1, threshold=0){
  cpt=c(0,0)
  delta=c(0,0)
  Delta=0

  nom_vecteur1 <- deparse(substitute(vecteur1))
  nom_vecteur2 <- deparse(substitute(vecteur2))
  if (grepl("[1]$" , nom_vecteur1) && grepl("[1]$" , nom_vecteur2)) { # tte/tte
    for(i in 1:length(vecteur1[[1]])){
      for (j in 1:length(vecteur2[[1]])){

        if(vecteur1[[2]][[i]]==1 && vecteur2[[2]][[j]]==1){

          if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
          else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
          else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
        }
      }
    }
  }
}

```

```

if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==1){

  if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
  else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
  else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]}
}

if(vecteur1[[2]][[i]] ==1 && vecteur2[[2]][[j]] ==0){

  if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
  else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
  else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
}

if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==0){

  if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
  else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
  else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]}
}
}
}

for(i in 1:length(vecteur1[[1]])){
  for (j in 1:length(vecteur2[[1]])){

    if(vecteur1[[4]][[i]]==1 && vecteur2[[4]][[j]]==1){

      if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] > threshold){cpt[2]=cpt[2]+1}
      else if(abs(vecteur1[[2]][[i]] - vecteur2[[2]][[j]]) <= threshold){cpt[2]=cpt[2]}
      else if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] < - threshold){cpt[2]=cpt[2]-1}
    }

    if(vecteur1[[4]][[i]] ==0 && vecteur2[[4]][[j]] ==1){

      if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] > threshold){cpt[2]=cpt[2]+1}
      else if(abs(vecteur1[[2]][[i]] - vecteur2[[2]][[j]]) <= threshold){cpt[2]=cpt[2]}
      else if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] < - threshold){cpt[2]=cpt[2]}
    }

    if(vecteur1[[4]][[i]] ==1 && vecteur2[[4]][[j]] ==0){

      if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] > threshold){cpt[2]=cpt[2]}
      else if(abs(vecteur1[[2]][[i]] - vecteur2[[2]][[j]]) <= threshold){cpt[2]=cpt[2]}
      else if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] < - threshold){cpt[2]=cpt[2]-1}
    }

    if(vecteur1[[4]][[i]] ==0 && vecteur2[[4]][[j]] ==0){

      if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] > threshold){cpt[2]=cpt[2]}
      else if(abs(vecteur1[[2]][[i]] - vecteur2[[2]][[j]]) <= threshold){cpt[2]=cpt[2]}
      else if(vecteur1[[2]][[i]] - vecteur2[[2]][[j]] < - threshold){cpt[2]=cpt[2]}
    }
  }
}

```

```

    }

    }
}
print(cpt)
}

if (grepl("[2]$", nom_vecteur1) && grepl("[2]$", nom_vecteur2)) { # tte/continue
  for(i in 1:length(vecteur1[,2])){
    for (j in 1:length(vecteur2[,2])){

      if(vecteur1[[2]][[i]] ==1 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

      if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]}
      }

      if(vecteur1[[2]][[i]] ==1 && vecteur2[[2]][[j]] ==0){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

      if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]}
      }

    }
  }

  for(i in 1:length(vecteur1[,2])){
    for (j in 1:length(vecteur2[,2])){

      if(vecteur1[,2][[i]] - vecteur2[,2][[j]] > threshold){
        cpt[2] = cpt[2] + 1
      } else if (vecteur1[,2][[i]] - vecteur2[,2][[j]] < - threshold){
        cpt[2] = cpt[2] - 1
      }
      else if (abs(vecteur1[,2][[i]] - vecteur2[,2][[j]])<= threshold){cpt[2]=cpt[2]}
    }
  }
}

```

```

    }
    }
    print(cpt)

}

if (grepl("[3]$" , nom_vecteur1) && grepl("[3]$" , nom_vecteur2)) {# tte/binaire
  for(i in 1:length(vecteur1[,2])){
    for (j in 1:length(vecteur2[,2])){

      if(vecteur1[[2]][[i]] ==1 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

      if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]+1}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

      if(vecteur1[[2]][[i]] ==1 && vecteur2[[2]][[j]] ==0){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

      if(vecteur1[[2]][[i]] ==0 && vecteur2[[2]][[j]] ==1){

        if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] > threshold){cpt[1]=cpt[1]}
        else if(abs(vecteur1[[1]][[i]] - vecteur2[[1]][[j]]) <= threshold){cpt[1]=cpt[1]}
        else if(vecteur1[[1]][[i]] - vecteur2[[1]][[j]] < - threshold){cpt[1]=cpt[1]-1}
      }

    }
  }

  for(i in 1:length(vecteur1[,2])){
    for (j in 1:length(vecteur2[,2])){

      if(vecteur1[,2][[i]] ==1 && vecteur2[,2][[j]]==0){
        cpt[2] = cpt[2] + 1
      } else if (vecteur1[,2][[i]] ==0 && vecteur2[,2][[j]]==1){
        cpt[2] = cpt[2] - 1
      }
      else {cpt[2]=cpt[2]}
    }
  }
  print(cpt)
}

```

```

}

if (grepl("[4]$", nom_vecteur1) && grepl("[4]$", nom_vecteur2)) { # continue / continue
  for(l in 1:2){
    for(i in 1:length(vecteur1[,l])){
      for (j in 1:length(vecteur2[,l])){

        if(vecteur1[i,l] - vecteur2[j,l] > threshold){
          cpt[l] = cpt[l] + 1
        } else if (vecteur1[i,l] - vecteur2[j,l] < - threshold){
          cpt[l] = cpt[l] - 1
        }

        else if (abs(vecteur1[i,l]-vecteur2[j,l])<= threshold){cpt[l]=cpt[l]}
      }
    }
  }
  print(cpt)
}

if (grepl("[5]$" , nom_vecteur1) && grepl("[5]$" , nom_vecteur2)) { # continue / binaire
  for(i in 1:length(vecteur1[,1])){
    for (j in 1:length(vecteur2[,1])){

      if(vecteur1[i,1] - vecteur2[j,1] > threshold){
        cpt[1] = cpt[1] + 1
      } else if (vecteur1[i,1] - vecteur2[j,1] < - threshold){
        cpt[1] = cpt[1] - 1
      }

      else if (abs(vecteur1[i,1]-vecteur2[j,1])<= threshold){cpt[1]=cpt[1]}
    }
  }
  for(i in 1:length(vecteur1[,2])){
    for (j in 1:length(vecteur2[,2])){

      if(vecteur1[i,2]==1 && vecteur2[,2][[i]]==0){
        cpt[2] = cpt[2] + 1
      } else if (vecteur1[i,2]==0 && vecteur2[,2][[i]]==1){
        cpt[2] = cpt[2] - 1
      }

      else {cpt[2]=cpt[2]}
    }
  }
  print(cpt)
}

if (grepl("[6]$", nom_vecteur1) && grepl("[6]$", nom_vecteur2)) { # binaire / binaire
  for(l in 1:2){
    for(i in 1:length(vecteur1[,l])){
      for (j in 1:length(vecteur2[,l])){

        if(vecteur1[i,l]==1 && vecteur2[,l][[i]]==0){

```

```

        cpt[1] = cpt[1] + 1
    } else if (vecteur1[i,1]==0 && vecteur2[,1][[i]]==1){
        cpt[1] = cpt[1] - 1
    }
    else {cpt[1]=cpt[1]}
}
}
}
print(cpt)
}
delta[1]=(cpt[1])/(n*n)
delta[2]=(cpt[2])/(n*n)
Delta=(delta[1]+w*delta[2])

return(c(cpt,delta,Delta))
}

```

```
## [1] 10526 -6966
```

```
## [1] 8828 28870
```

```
## [1] 17554 11000
```

```
## [1] 35368 26242
```

```
## [1] 20064 14000
```

```
## [1] 12000 7000
```

```
##      Cas.de.figure Valeur.de.Delta Valeur.de.delta.1 Valeur.de.delta.2.
## 1      tte/tte      0.08900      0.26315      -0.17415
## 2      tte/continue 0.94245      0.22070      0.72175
## 3      tte/binaire  0.71385      0.43885      0.27500
## 4 continue/continue 1.54025      0.88420      0.65605
## 5 continue/binaire  0.85160      0.50160      0.35000
## 6 binaire/binaire   0.47500      0.30000      0.17500
## Valeur.de.p_ij.1. Valeur.de.p_ij.2.
## 1      10526      -6966
## 2      8828      28870
## 3      17554      11000
## 4      35368      26242
## 5      20064      14000
## 6      12000      7000
```

Win ratio