

# Iris Example

---

## Purpose

To demonstrate that for the Iris dataset our *Mathematica* package `nnc` gives the same results as the R package. This is just an example check -- many other checks were also carried out.

---

## Example Check

```
In[1]:= Needs["nnc`"]
```

Nearest Neighbor Autocovariates.

July 30, 2010

```
In[2]:= X = Import["StatisticsExamples/Data/iris.dat"];
```

```
In[3]:= y = Flatten[Table[{"setosa", "versicolor", "virginica"}, {50}] // Transpose];
```

```
In[8]:= a = NNC[X, y, 17];
```

```
      i = Range[146, 150];
```

```
      TableForm[Drop[a, 145], TableHeadings -> {Range[146, 150], None}]
```

Out[10]/TableForm=

146	0.0588235	0.941176
147	0.235294	0.764706
148	0.0588235	0.941176
149	0.	1.
150	0.176471	0.823529

## ■ Comparison With R

---

```
> library(nnc)
```

---

```
Loading required package: class
```

```
Loading required package: nnet
```

---

---

```
> library(RWinEdt)
> options(prompt="R> ")
R> k<-17
R> X<-iris[, 1:4]
R> Y<-iris[, 5]
R> n<-length(Y)
R> y<-numeric(n)
R> classes <- unique(Y)
R> Q <- length(classes)
R> ind1 <- Y==classes[1]
R> y[ind1] <- -1
R> z <- matrix(numeric(n*(Q-1)), nrow=n)
R> for (j in 2:Q){
+   indk <- Y==classes[j]
+   indOther <- !(ind1|indk)
+   y[indk] <- 1
+   y[indOther] <- -1
+   zA <- nnc(X=X, Y=y, k=k)
+   y[indOther] <- 1
+   zB <- nnc(X, y, k)
+   z[,j-1] <- (zA+zB)/2
+ }
R> tail(z,5)
```

---

	[,1]	[,2]
[146,]	0.05882353	0.9411765
[147,]	0.23529412	0.7647059
[148,]	0.05555556	0.9444444
[149,]	0.00000000	1.0000000
[150,]	0.17647059	0.8235294

---