MGC

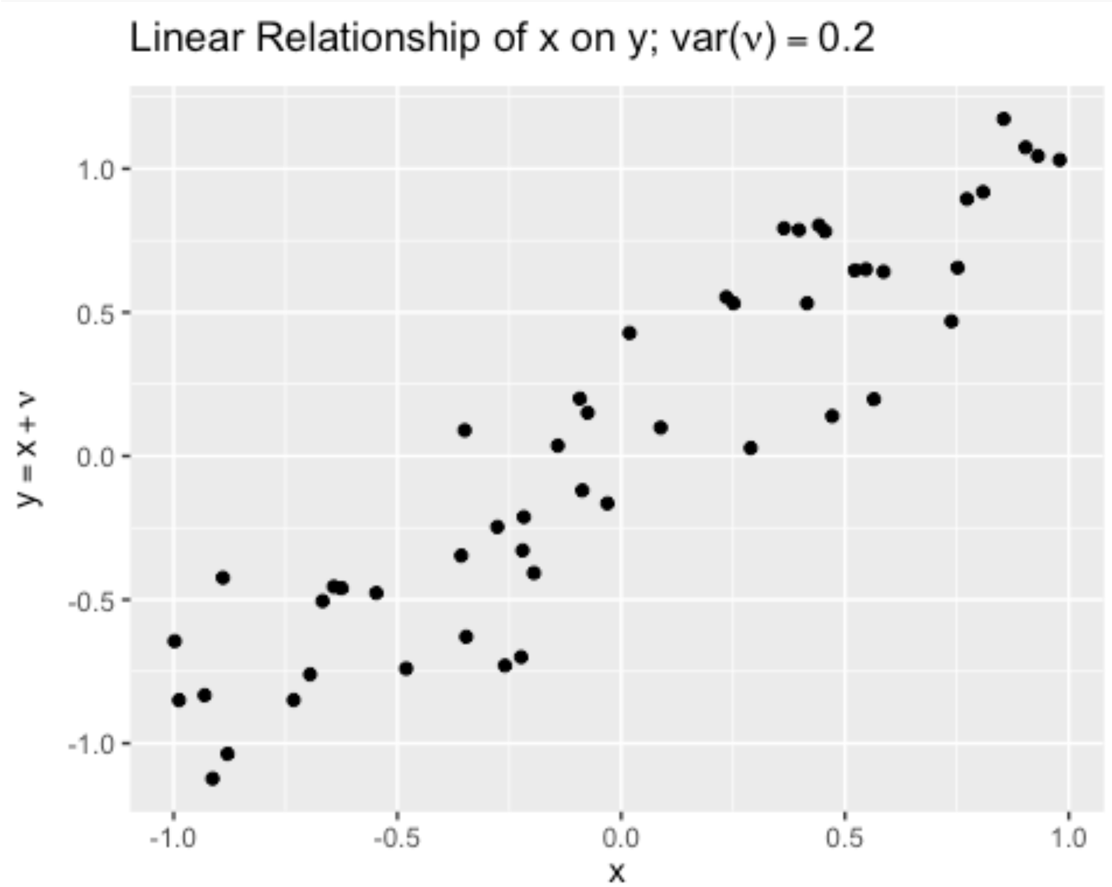
The x data provided is by sampling 100 times from a uniform distribution on the interval [-1, 1], and the y data is formed by adding normally distributed error with variance 0.2 (indicating a linear relationship). The resulting MGC value suggests a strong linear relationship.

```r
require(devtools)

## Loading required package: devtools

install_github('neurodata/mgc', build_vignettes=TRUE)

## Skipping install of 'mgc' from a github remote, the SHA1 (835d139b) has no
t changed since last install.
##   Use `force = TRUE` to force installation

require(mgc)

## Loading required package: mgc

## Loading required package: SDMTools

vignette("MGC", package="mgc")

## starting httpd help server ...

##  done

require(mgc)
set.seed(12345)
mgc.sample(mgc::test_xunif, mgc::test_ylin)$statMGC

## [1] 0.891153
```

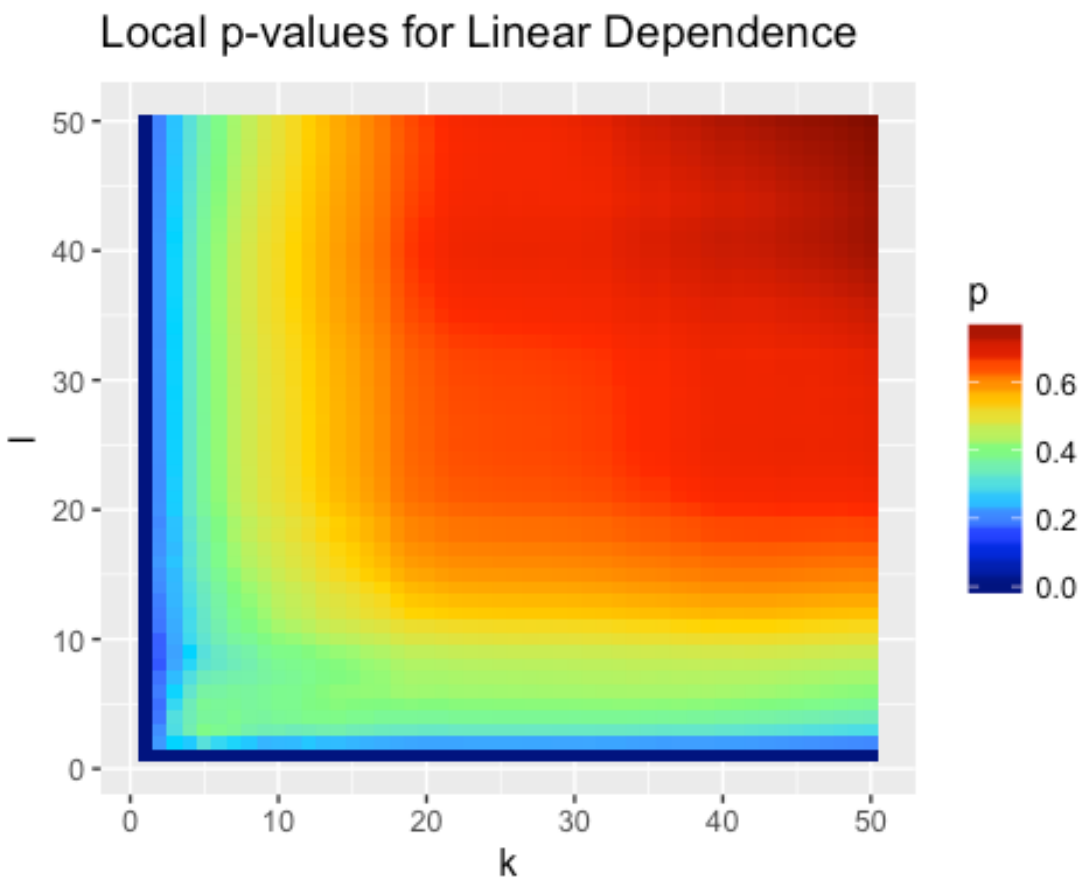
Below is a more in depth version of the previous code chunk using simulated data. Resulting MGC value also supports liner relationship presented in plot

```r
library(mgc)
require(ggplot2)

## Loading required package: ggplot2

require(latex2exp)

## Loading required package: latex2exp

n = 50
d = 1
eps = 0.2
x = runif(n, min=-1, max=1)
```

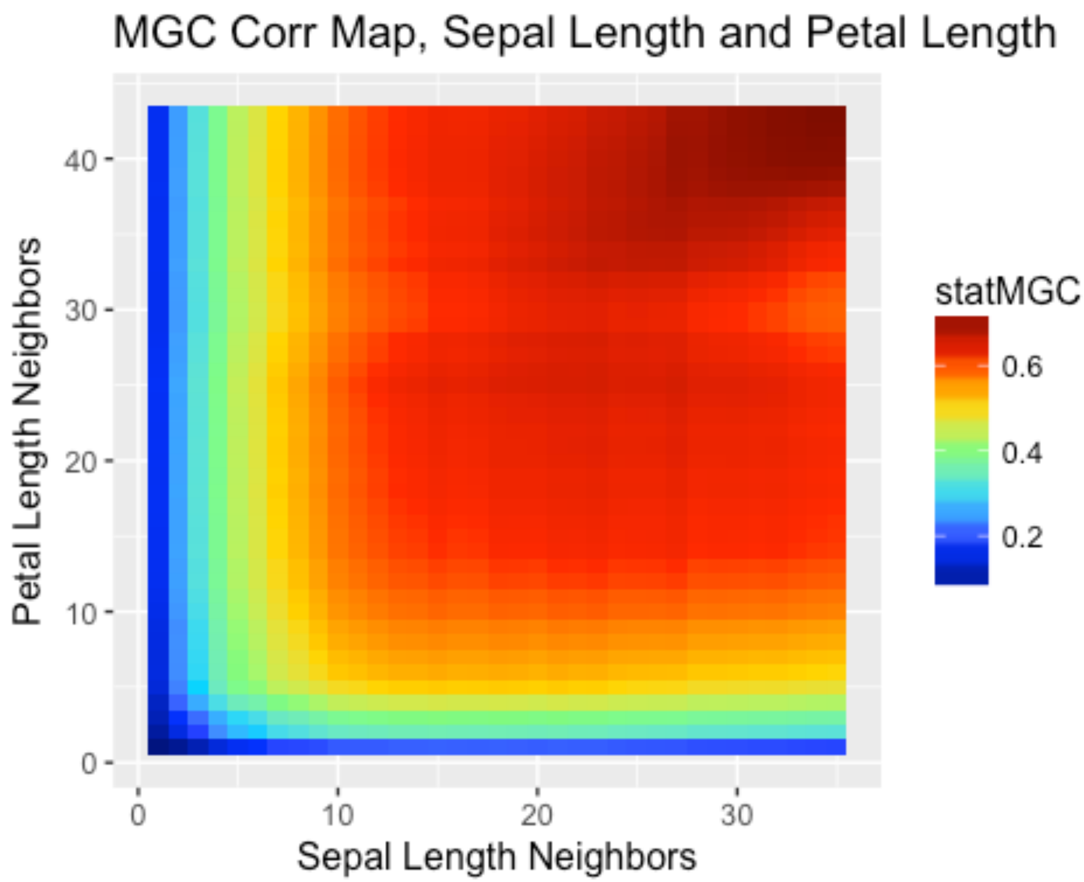
```r
A=matrix(0,d,1)
for (i in (1:d)){
  A[i]=1/i
}
nu = eps*rnorm(n, mean=0, sd=1)
y = x%*%A + nu

dat = data.frame(x=x, y=y)

ggplot(dat, aes(x=x, y=y)) +
  geom_point() +
  xlab(TeX("$x$")) +
  ylab(TeX("$y = x + \\nu$")) +
  ggtitle(TeX("Linear Relationship of $x$ on $y$; $var(\\nu) = 0.2$"))
```




```r
Xd = dist(x, diag=TRUE, upper=TRUE)
Yd = dist(y, diag=TRUE, upper=TRUE)
res <- mgc.test(Xd, Yd, rep=100)
mgc.plot.plot_matrix(res$localCorr, xlabel=TeX("k"), ylabel=TeX("l"), title =
TeX("Local $p$-values for Linear Dependence"), legend.name=TeX("p"))
```




```
print(res$optimalScale)

## [1] 2500

print(res$statMGC)

## [1] 0.7981858

#Real IRIS Data
set.seed(12345)
res <- mgc.sample(iris[,1], iris[,3])
mgc.plot.plot_matrix(res$localCorr, title="MGC Corr Map, Sepal Length and Pet
al Length", xlab="Sepal Length Neighbors", ylab="Petal Length Neighbors", leg
end.name = "statMGC")
```




```
print(res$statMGC)
```

```
## [1] 0.7337225
```

The next few demos show examples of discriminabilty.

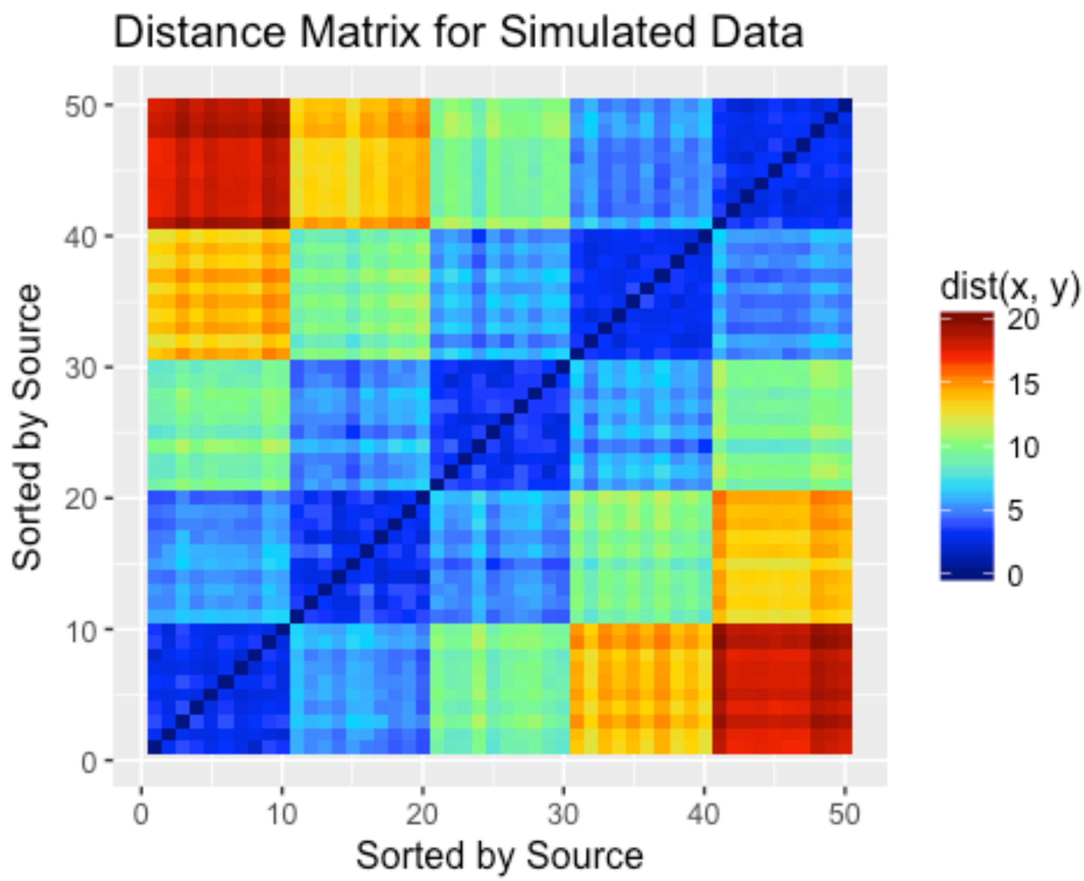
```
library(mgc)

nsrc <- 5
nobs <- 10
d <- 20
src_id <- array(1:nsrc)
set.seed(12345)
labs <- sample(rep(src_id, nobs))
dat <- t(sapply(labs, function(lab) rnorm(d, mean=lab, sd=0.5)))
discr.stat(dat, labs)
```

```
## [1] 0.9983889
```

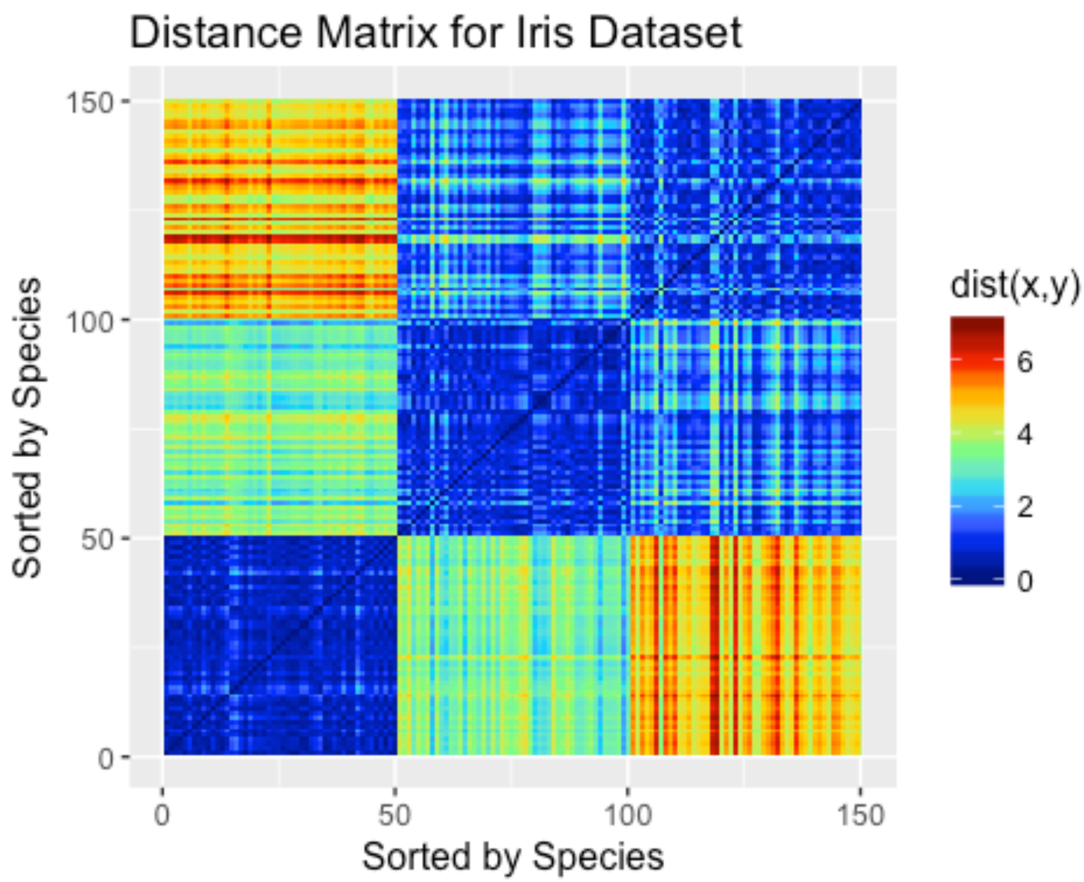
Shows that objects from the same source have a lower distance than objects from a different source.

```
vignette("Discriminability", package="mgc")
Dx <- as.matrix(dist(dat[sort(labs, index=TRUE)$ix,]), method='euclidian')
```

```
mgc.plot.plot_matrix(Dx, xlab="Sorted by Source", ylab="Sorted by Source", ti
tle="Distance Matrix for Simulated Data", legend.name="dist(x, y)")
```




```
discr.stat(Dx, sort(labs))

## [1] 0.9983889

#Real IRIS Data
Dx <- as.matrix(dist(iris[sort(as.vector(iris$Species), index=TRUE)$ix,c(1,2,
3,4)]))
mgc.plot.plot_matrix(Dx, xlab="Sorted by Species", ylab="Sorted by Species",
title="Distance Matrix for Iris Dataset", legend.name=TeX("$dist(x, y)$"))
```




```
discr.stat(iris[,c(1,2,3,4)], as.vector(iris$Species))

## [1] 0.9320476
```