THE UNIVERSITY OF MINNESOTA

Statistics 5401 September 17, 2005

Chi-Squared Q-Q plots to Assess Multivariate Normality

Suppose $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is a random sample from a p-dimensional multivariate distribution with population (true) mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Let $d_j^2 \equiv (\mathbf{x}_j - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_j - \boldsymbol{\mu})$, j = 1,...,n, be the generalized squared distances of the data points from $\boldsymbol{\mu}$. The quantities $\{d_1^2, d_2^2, \ldots, d_n^2\}$ are independent and all have the same distribution so they constitute a random. You can use them to assess the multivariate normality of $\mathbf{x}$.

When $\mathbf{x}$ is $N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$ (p-dimensional multivariate normal with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$), $d^2 \equiv (\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ has the χ_p^2 distribution (chi-squared on p-degrees of freedom).

Putting these together, you can conclude that, when the $\mathbf{x}_j$'s are a random sample from $N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$, $d_1^2, d_2^2, \ldots, d_n^2$ are a random sample from a χ_p^2 distribution.

Suppose you know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Then you can test H_0: "$\mathbf{x}$ is multivariate normal" by any test of the goodness-of-fit of $\{d_j^2\}$ to the χ_p^2 distribution, that is a test of H_0: $d^2 \equiv (\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ is χ_p^2. If the sample of d_j^2's fails such a test, that is you reject H_0, then you must also reject the null hypothesis you're really interested in, namely H_0: $\mathbf{x}$ is $N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$. However, if the test fails to reject, this does not necessarily imply that $\mathbf{x}$ is not $N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$.

A chi-squared *Q-Q plot* is one useful way informally to assess whether d^2 is distributed as χ_p^2. It is similar to a *normal scores plot* that is often used to assess *univariate* normality. It consists of two steps:

(a) Order the calculated d_j^2's in increasing order $d_{(1)}^2 < d_{(2)}^2 < \ldots < d_{(n)}^2$ (parenthesized subscripts are a standard notation to indicate that values are ordered). In MacAnova, you can order the d_j^2's using `sort()`.

(b) Plot the $d_{(j)}^2$'s against the chi-squared probability points $\chi_p^2(1-q_j)$, j = 1,2,...,n, where the q_j are equally spaced probabilities between 0 and 1, say $q_j = (j-.5)/n$, j = 1, 2,..., n. Here $\chi_p^2(\alpha)$ is the *upper* α-th probability point of χ_p^2 (chi-squared) on p degrees of freedom. You could also use $q_j = j/(n+1)$ spaced by $1/(n+1)$ on the probability side, but for consistency I will use $q_j = (j-.5)/n$, spaced by $1/n$.

In MacAnova you can compute $q_1, q_2, \ldots, q_n$ by `invchi((run(n)-.5)/n,p)`. Because the $d_{(j)}^2$'s are ordered, a Q-Q plot always increases to the right. If the data are multivariate normal and d^2 is in fact χ_p^2, the plot should be approximately a straight line through the origin with slope 1.

You should *always* include the origin (0,0) in the plot. You do this by including `xmin:0, ymin:0` as arguments to the plotting command

In most cases, a plot of $d_{(j)} = \sqrt{\{d_{(j)}^2\}}$ against $\sqrt{\{\chi_p^2(1-q_j)\}}$ is preferable since there is less piling up

of points at the lower end. This also should be a straight line through the origin with slope 1 and its straightness is usually easier to judge than the plot of $d_{(j)}^2$.

This would be straightforward if you did know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Unfortunately, except in rare cases, you *don't* know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and can't compute $d_{(j)}^2$. However, you can estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by $\hat{\mu} = \bar{x}$ and $\hat{\Sigma} = \mathbf{S}$, where $\mathbf{S}$ is the unbiased estimate of $\boldsymbol{\Sigma}$,. You can then compute $\hat{d}_{(1)}^2 \leq \hat{d}_{(2)}^2 \leq \ldots \leq \hat{d}_{(n)}^2$, where the $\hat{d}_{(j)}^2$ are the ordered values of the *estimated* squared generalized distances $\hat{d}_j^2 = (\mathbf{x}_j - \bar{x})'\mathbf{S}^{-1}(\mathbf{x}_j - \bar{x})$.

Although the $\hat{d}_j^2$'s are not distributed *exactly* as χ_p^2 under the null hypotheses of multivariate normality, and are not fully independent, a χ_p^2 Q-Q plot or a $\sqrt{(\chi_p^2)}$ Q-Q plot based on them should still be approximately linear when $\mathbf{x}$ is N_p, at least when n is not too small.

When $\mathbf{x}$ in multivariate normal, so is any subset of variables. So you can sometimes get further insight by testing the multivariate normality of one or more subsets of q < p of the variables in $\mathbf{x}$. If q > 1 you can make χ_q^2 or $\sqrt{(\chi_q^2)}$ Q-Q plots. If q = 1, you can assess marginal univariate normality by making a **normal scores plot**, computing normal scores by MacAnova function `rankits()`.

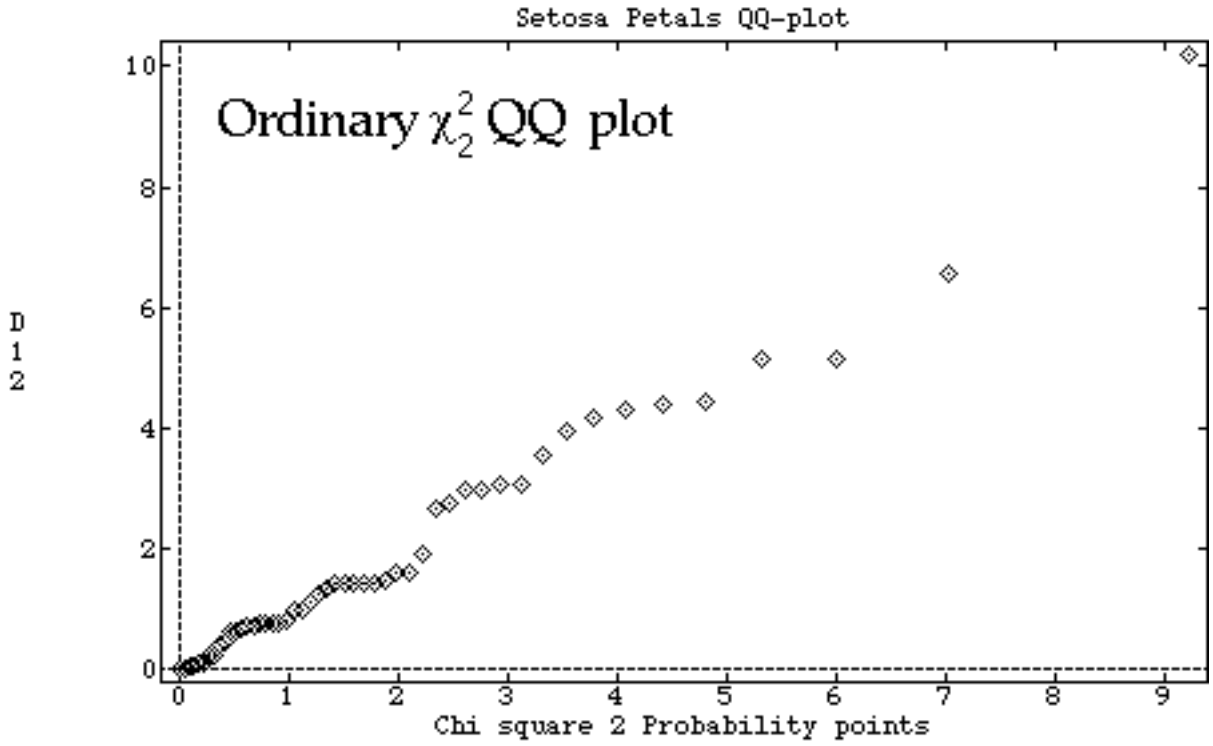
When your analysis involves a **multivariate regression** (p > 1 dependent variables) or **multivariate analysis of variance** (MANOVA), you can assess normality by any of these procedures applied to the *residuals* from the model fit. χ^2 Q-Q plots of residuals generalize to the multivariate case the common use of normal scores plots of univariate residuals.

The following MacAnova output illustrates the use of a Q-Q plot to examine the multivariate normality of the Fisher iris data from Table 11.5 on p. 566 of Johnson & Wichern. These consist of four measurements, x_1 = sepal width, x_2 = sepal length, x_3 = petal width, and x_4 = petal length, on 50 flowers from each of three varieties of iris, *I. setosa*, *I. versacolor*, and *I. virginica*. The MacAnova session makes use of macro `distcomp()` in the standard macro file Mulvar.mac. `distcomp()`.

```
Cmd> y <- read("","t11_05")  #read from JWData5.txt
) Data from Table 11.5 p. 657-658 in
) Applied Mulivariate Statistical Analysis, 5th Edition
) by Richard A. Johnson and Dean W. Wichern, Prentice Hall, 2002
) These data were edited from file T11-5.DAT on disk from book
) The variety number was moved to column 1
) Measurements on petals of 4 varieties of Iris. Originally published
in
) R. A. Fisher, The use of mltiple measurements in taxonomic problems,
) Annals of Eugenics, 7 (1936) 179-198
```

```
) Col. 1: variety number (1 = I. setosa, 2 = I. versicolor,
)                         3 = I. virginica)
) Col. 2: x1 = sepal length
) Col. 3: x2 = sepal width
) Col. 4: x3 = petal length
) Col. 5: x4 = petal width
) Rows 1-50:    group 1 = Iris setosa
) Rows 51-100:  group 2 = Iris versicolor in
) Rows 101-150: group 3 = Iris virginica in
Read from file "TP1:Stat5401:Stat5401F04:Data:JWData5.txt"

Cmd> varieties <- y[,1]

Cmd> setosa <- y[varieties==1,-1] # last 4 cols for variety 1

Cmd> dim(setosa) # dimensions
(1)           50            4

Cmd> usage(distcomp)
distcomp(y), REAL matrix y with no MISSING values

Cmd> d12 <- distcomp(setosa[,vector(1,2)])# distances based on x1, x2

Cmd> n <- nrows(setosa) # number cases is 1st dimension of setosa

Cmd> q <- ncols(setosa) # 2

Cmd> x <- invchi((run(n)-.5)/n,q) # chi-squared prob points
```

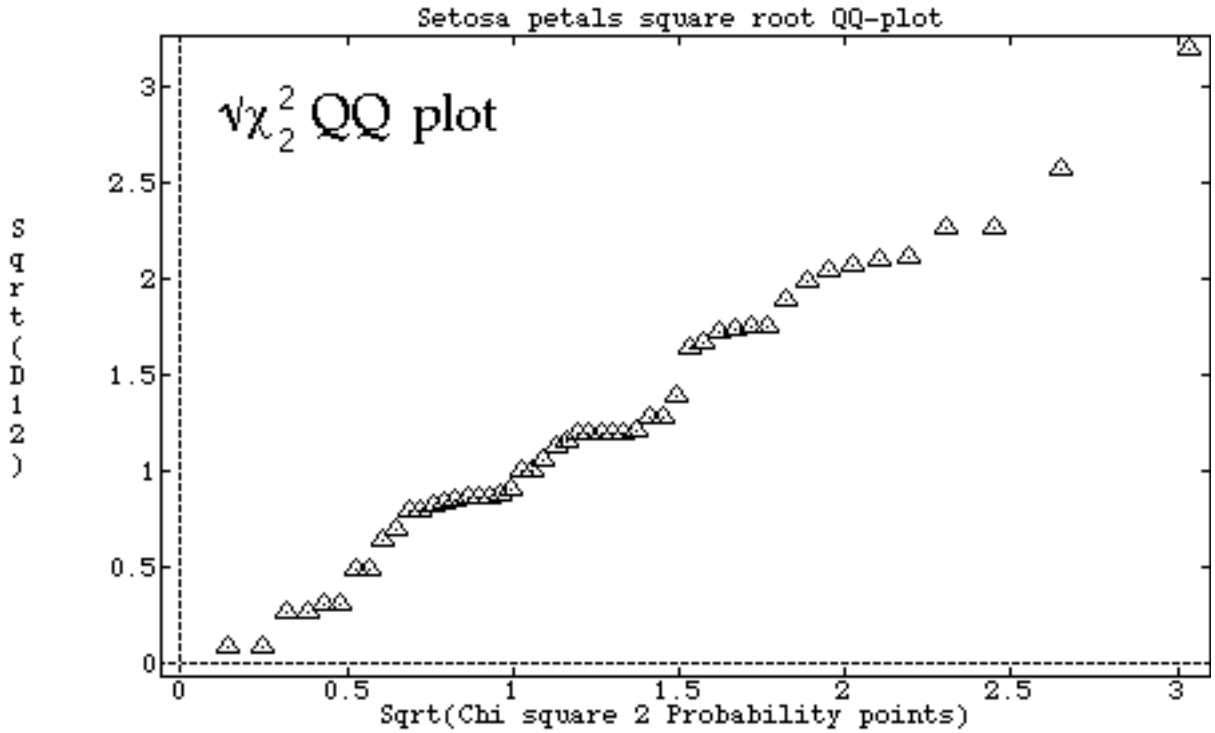
```
Cmd> # Now make a plot make plot with diamond symbol
Cmd> # Characters like "\1","\2","\3","\4","\5", "\6", "\7",
Cmd> # give diamond, plus, square, cross, triangle, asterisk, dot
Cmd> plot(x,D12:sort(d12),symbols:"\1",xmin:0,ymin:0,\
title:"Setosa Petals QQ-plot",xlab:"Chi square 2 Probability points")
```




```
Cmd> # Square root gamma plot is often easer to see patterns in
```

Note the use of `xmin:0,ymin:0` to ensure that the point (0,0) is in the plot.

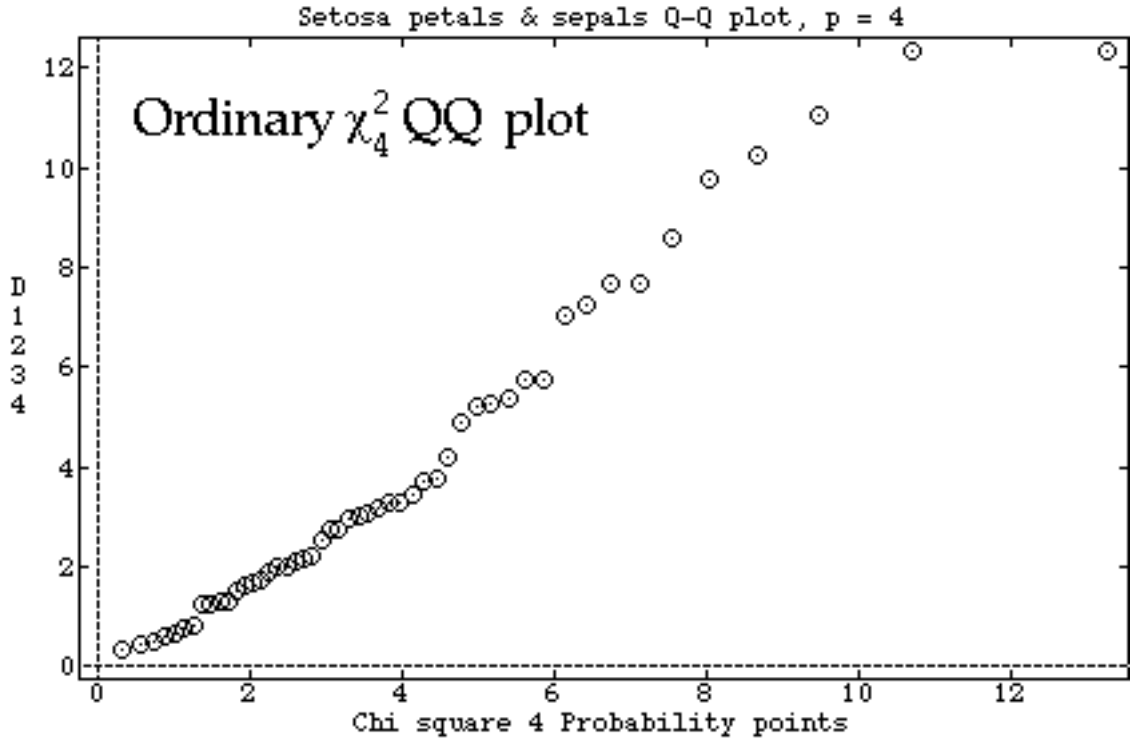
```
Cmd> plot(sqrt(x),sqrt(sort(d12)),symbols:"\5",ylab:"Sqrt(D12)",\
    xlab:"Sqrt(Chi square 2 Probability points)",\
    title:"Setosa petals square root QQ-plot",xmin:0,ymin:0)
```



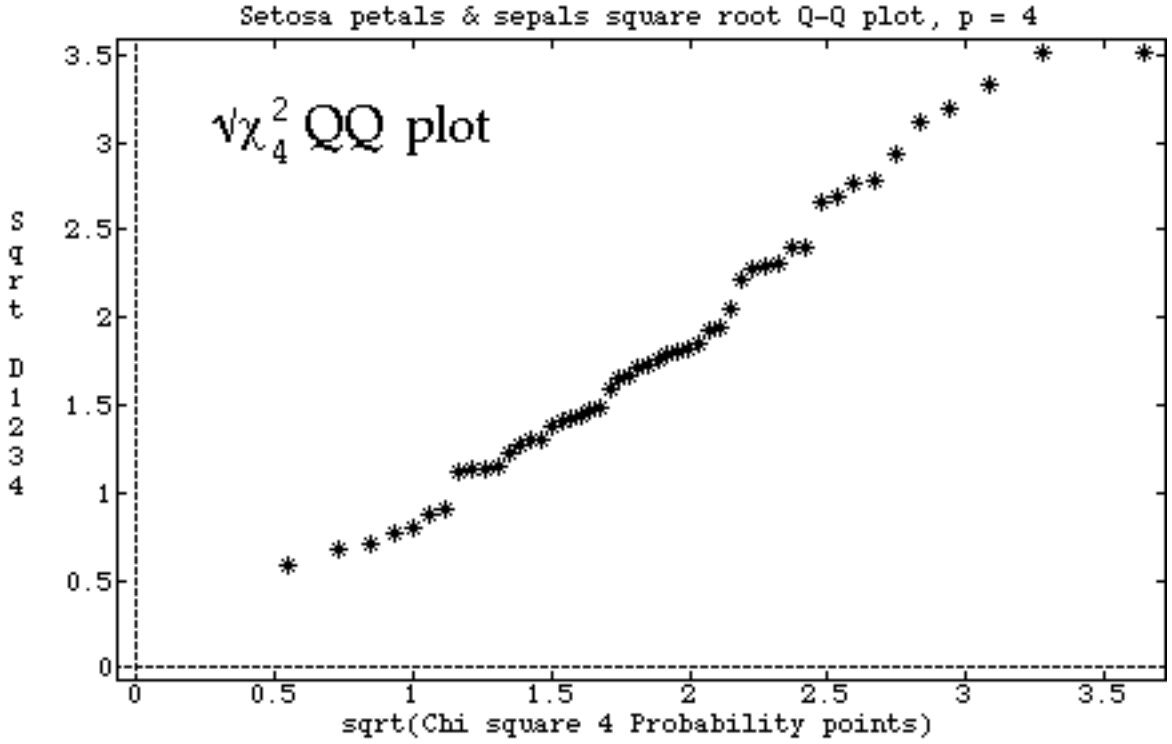
Plotting square roots avoids the crowding of points at the lower end so you can see better what is going on.

Now do the same using all four variables.

```
Cmd> d1234 <- distcomp(setosa) # distances based on x1, x2, x3, x4
Cmd> p <- ncols(setosa); x <- invchi((run(n)-.5)/n,p) # p = 4
Cmd> plot(x,D1234:sort(d1234),symbols:"\10",\
       title:"Setosa petals & sepals Q-Q plot, p = 4",\
       xlab:"Chi square 4 Probability points",xmin:0,ymin:0)
```



```
Cmd> # Now make square root gamma plot using asterisk
Cmd> plot(sqrt(x),sqrt(sort(d1234)),symbols:"\6",\
       title:"Setosa petals & sepals square root Q-Q plot, p = 4",\
       xlab:"sqrt(Chi square 4 Probability points)",\
       ylab:"Sqrt D1234",xmin:0,ymin:0)
```



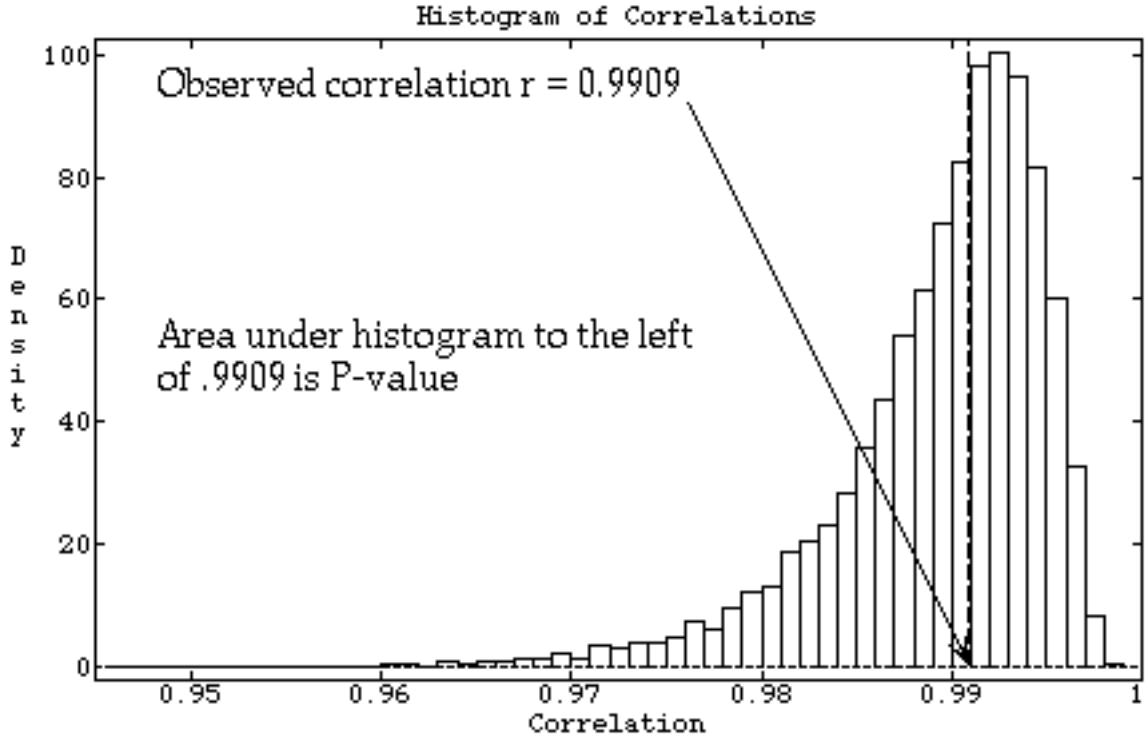
Examining the Q-Q plot does not constitute a true significance test. However, you can base a formal significance test on it. By analogy with the correlation test of univariate normality (a close relative of the Wilk-Shapiro test), a possible test is the correlation r between the ordered probability points (horizontal axis in the plots) and the ordered distances (vertical axis int he plots. You reject normality when r is small enough since this indicates departure from a straight line.

```
Cmd> r <- cor(sort(sqrt(d1234)),sqrt(x))[1,2]; r
(1,1)      0.99086
```

This seems pretty high and thus possibly non-significant, but critical values or a P-value you can't tell that it's not significantly low. However, you can use simulation to *estimate* the P-value. Using a computer, you can (a) generate M multivariate normal samples, where M is large and (b) compute r ffrom each sample. You can then estimate the P-value by computing the proportion that are smaller than .99086 which estimates the the probability of a value smaller than .99086. Here's how to do it in MacAnova.

```
Cmd> M <- 10000;R <- rep(0,M) # place to put simulated r's
Cmd> for(i,1,M){ # compute M correlations
    R[i] <- cor(sqrt(sort(distcomp(matrix(rnorm(n*p),n)))),\
            sqrt(x))[1,2];;}
Cmd> min(R)# minimum
(1)      0.94608
```

```
Cmd> hist(R,vector(.94,.001),\
    title:"Histogram of Correlations",xlab:"Correlation", show:F)
Cmd> addlines(rep(r,2),vector(0,110),linetype:2) #line at observed r
```



The dashed line marks the observed value .9909. You can compute an estimated P-value by

```
Cmd> sum(R <= r)/M # estimated P-value
(1,1)       0.5102
```

This shows no evidence of non-normality. `sum(R <= r)` counts the number of elements of `R` less than or equal to the observed value.

Incidentally, since the simulation used exactly multivariate normal data, this does not assume that the distances are a random sample from χ_p^2.

Also, although the simulation generated multivariate normal data with population variance matrix $\mathbf{I}_p$, there is no loss of generality. From a multivariate normal vector $\mathbf{x}$ with variance matrix $\mathbf{I}_p$ you can generate multivariate vector $\mathbf{y}$ with any covariance matrix $\mathbf{\Sigma}$ as $\mathbf{y} = \mathbf{A}'\mathbf{x}$ where $\mathbf{A}$ satisfies $\mathbf{A}'\mathbf{A} = \mathbf{\Sigma}$, and it is always possible to find such a matrix $\mathbf{A}$. But the distances computed from $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ are identical to those computed from $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$. Thus the distribution of the distances does not depend on $\mathbf{\Sigma}$.