Displays for Statistics 5401/8401

Lecture 7

September 21, 2005

Christopher Bingham, Instructor

612-625-1024, kb@umn.edu
372 Ford Hall

Class Web Page

http://www.stat.umn.edu/~kb/classes/5401

Multistandardizing with $\boldsymbol{\Sigma}^{1/2}$

A matrix square root $\boldsymbol{\Sigma}^{1/2}$ of a positive definite symmetric matrix $\boldsymbol{\Sigma}$ satisfies

$$(\boldsymbol{\Sigma}^{1/2})'(\boldsymbol{\Sigma}^{1/2}) = \boldsymbol{\Sigma}$$

Since $\boldsymbol{\Sigma}^{-1} = (\boldsymbol{\Sigma}^{1/2})^{-1}((\boldsymbol{\Sigma}^{1/2})')^{-1}$, a matrix square root of $\boldsymbol{\Sigma}^{-1}$ is $((\boldsymbol{\Sigma}^{1/2})^{-1})'$.

When $\mathbf{y}$ is a random vector with mean $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$, you can use $\boldsymbol{\Sigma}^{1/2}$ to multistandardize $\mathbf{y}$.

Define $\mathbf{A} = \boldsymbol{\Sigma}^{-1/2} \equiv (\boldsymbol{\Sigma}^{1/2})^{-1}$ and let $\mathbf{z} \equiv \mathbf{A}'(\mathbf{y} - \boldsymbol{\mu})$.

Then

$$\begin{aligned} V[\mathbf{z}] &= ((\boldsymbol{\Sigma}^{1/2})^{-1})'\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{1/2})^{-1} \\ &= ((\boldsymbol{\Sigma}^{1/2})')^{-1}(\boldsymbol{\Sigma}^{1/2})'(\boldsymbol{\Sigma}^{1/2})(\boldsymbol{\Sigma}^{1/2})^{-1} \\ &= \mathbf{I}_p\mathbf{I}_p = \mathbf{I}_p. \end{aligned}$$

Since $E[\mathbf{z}] = \mathbf{0}$ and $V[\mathbf{z}] = \mathbf{I}_p$, $\mathbf{z}$ is a *multistandardized version* of $\mathbf{y}$.

To multistandardize a n by p data matrix $\mathbf{Y}$, you use $(\mathbf{S}^{1/2})^{-1}$:

$$\tilde{\mathbf{Y}} = (\mathbf{Y} - \mathbf{1}_n\overline{\mathbf{y}})\,(\mathbf{S}^{1/2})^{-1}$$

This transforms the data $\mathbf{y}_i$ for case i to

$$\tilde{\mathbf{y}}_i = ((\mathbf{S}^{1/2})^{-1})'(\mathbf{y}_i - \overline{\mathbf{y}})$$

```
Cmd> data <- read("","T01_06") # Multiple sclerosis data
Cmd> # Column 1 is group number, 1 = non-MS, 2 = MS
Cmd> nonms <- data[data[,1] == 1,-1] # non-MS data
Cmd> ybar <- tabs(nonms,mean:T)
Cmd> s <- tabs(nonms,covar:T)
Cmd> sqrt_s <- cholesky(s) # triangular matrix square root
Cmd> newy <- (nonms - ybar') %/% sqrt_s
Cmd> tabs(newy,mean:T,covar:T)
component: mean
(1) -2.9606e-16  1.8681e-15 -3.1376e-16 -1.0364e-15  2.7997e-16
component: covar
(1,1)          1  9.2946e-17  1.0609e-16  1.3609e-16  9.3937e-17
(2,1) 9.2946e-17           1 -6.7996e-17    3.15e-17 -8.2594e-18
(3,1) 1.0609e-16 -6.7996e-17           1  -3.191e-18  1.4001e-16
(4,1) 1.3609e-16    3.15e-17  -3.191e-18           1 -1.6569e-17
(5,1) 9.3937e-17 -8.2594e-18  1.4001e-16 -1.6569e-17           1
```

Except for rounding error, the sample mean of **newy** is $\mathbf{0}$ and the sample variance matrix is $\mathbf{I}_5$.

Working as Advertised

All statistical procedures, including

- confidence intervals or regions
- hypothesis tests,

require certain assumptions to be true such as

- Data or errors are random sample
- The data or errors from a normal population
- Variance σ^2 or variance matrix $\mathbf{\Sigma}$ is constant.

Q: Why do you need such assumptions to be satisfied?

A: So that the procedures should "work as advertised" or "work as claimed."

What does this mean?

A significance or hypothesis test "works as advertised" when

actual type I error rate ($P(\text{reject}\,|\,H_0)$) = *intended* or *claimed* significance level α.

A confidence interval or region "works as advertised" when

actual confidence level = P(interval or region includes the true parameter) = *intended* or *claimed* confidence level.

For example, if a univariate sample X_1, ..., X_n is not random but $\text{corr}[X_i, X_{i+1}] = \rho \neq 0$, $V[\overline{X}] \cong (\sigma_x^2/n)(1 + \rho)$.

This means that in large samples,

$t = (\overline{X} - \mu)/(s_x/\sqrt{n}) \approx N(0, 1 + \rho)$, so

$P(|t| > z_{\alpha/2}) \cong P(|t| > z_{\alpha/2}/\sqrt{(1+\rho)}) \not\cong \alpha$.

So it's important to assess the truth of assumptions.

Assessing multivariate Normality

Many multivariate statistical procedures require multivariate normality in order to "**work as advertised**".

Thus it is important to assess the truth of null hypotheses like

$$H_0\text{: } \mathbf{X} \text{ is } N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Better yet is a formal significance test of H_0. This is a hard problem.

The simplest situation is when $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ is a *random sample* from some p-dimensional multivariate distribution with $E[\mathbf{X}] = \boldsymbol{\mu}$ and $V[\mathbf{X}] = \boldsymbol{\Sigma}$ and you want to determine if there is evidence the distribution is not normal.

Testing the goodness-of-fit to a multivariate normal is difficult, and virtually impossible with small samples.

Focus of most approaches

- Check whether the distribution of **X** appears not to have some particular property of the N_p distribution.
- When the distribution of **X** appears to not to have the property, you conclude **X** is not multivariate normal.

Even if **X** does satisfy the property, that is no guarantee it is normal.

Properties of Multivariate Normal

- Each individual variable is N_1.
  Every subset of q variables is N_q.
- $(X - \mu)'\Sigma^{-1}(X - \mu)$ distributed as χ_p^2
- **Linearity of regression** of each X_j on the other variables:
  $E[X_j \mid X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_p]$ is linear in $X_1,\ldots,X_{j-1}, X_{j+1},\ldots,X_p$
- **Constant conditional variances**
  $\sigma_{jj.12\ldots j-1,j+1\ldots p} = V[X_j \mid X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_p]$
  doesn't depend on $X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_p$, $j = 1,\ldots,p$

You can assess the two last two properties by standard multiple regression methods, and in particular by plots of residuals against fitted values.

The most common way to assess *univariate* normality (normality of a single variable) is a normal scores plot - a plot of

the order statistics $X_{(i)}$, the values in the sample arranged in order

$$X_{(1)} \le X_{(2)} \le \ldots \le X_{(n)}$$

against

"normal scores" or probability points a_i.

If there is too much curvature in the plot, there is evidence against normality.

MacAnova normal scores

rankits(n:N) and **rankits(run(N))** both compute normal scores by

invnor((run(N) - .375)/(N + .25)).

This differs from what the text suggests for normal scores, which is equivalent to

invnor((run(N) - .5)/N).

The difference is not important.

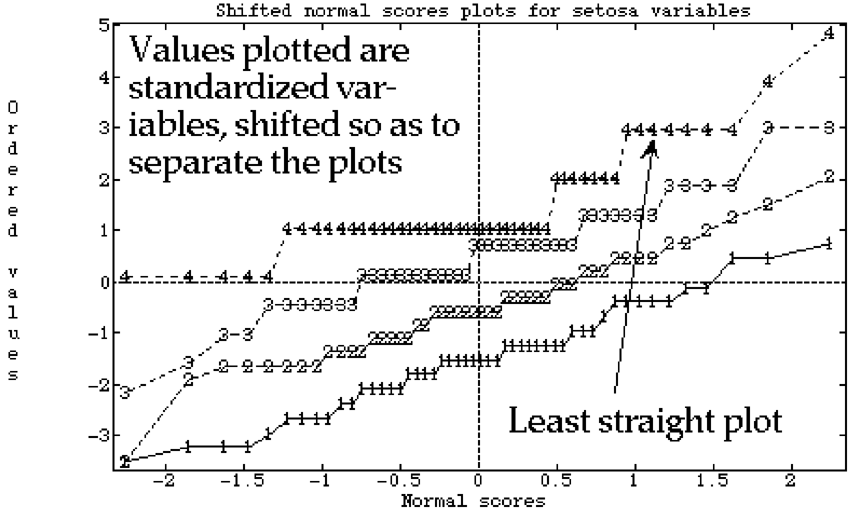
```
Cmd> irisdata <- matread("JWdata4.txt","jwt11-5")
) Data from Table 11.5 p. 657-658 in
) Applied Mulivariate Statistical Analysis, 5th Edition
) by Richard A. Johnson and Dean W. Wichern, Prentice Hall, 2002
) These data were edited from file T11-5.DAT on disk from book
) The variety number was moved to column 1
) Measurements on petals of 4 varieties of Iris.  Originally
published in
) R. A. Fisher, The use of multiple 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> groups <- irisdata[,1]; y <- irisdata[,-1]

Cmd> setosa <- y[groups==1,]

Cmd> z <- sort(standardize(setosa))
```

standarize(x) standardizes the columns of **x** so the vertical scales of normal scores plots will be comparable. If something further isn't done, the plots for the four variables will overlap.

```
Cmd> shiftedz <- z + (run(4) - 2.5)'
```

Adding **(run(4)-2.5)'** adds -1.5, -.5, .5, 1.5 to the 4 columns of standardized data. This will separate them in a plot.

```
Cmd> nscores <- rankits(n:nrows(setosa)) # normal scores
Cmd> lineplot(nscores, shiftedz, ylab:"Ordered values",\
     symbols:run(4),xlab:"Normal scores",\
  title:"Shifted normal scores plots for setosa variables")
```



Values plotted are standardized variables, shifted so as to separate the plots

Least straight plot

Normal scores plots of standardized data look the same as normal scores plots of the original data.

The only one that seems quite curved is the top plot, the one for variable 4.

Plots like these help *assess* normality but do not provide a significance test.

The most common test for *univariate* normality (normality of a single variable) is probably a statistic related to the Wilk-Shapiro test statistic, namely the correlation statistic

$$W = \widehat{\text{corr}}(X_{(i)}, a_i)$$

W is one way of measuring how straight the normal scores plot is. The more curvature in the plot, the lower W will be, although it will always be positive.

Thus in a test based on W, you reject for *small* values. That is, the test is a lower tail test.

Here I calculate all the correlations of the sorted data with the normal scores (rankits) in **nscores**.

```
  Cmd> w <- vector(cor(nscores,sort(setosa))[1,run(2,5)]); w
  (1)     0.99081     0.98188     0.97418     0.89172
```

cor(nscores,sort(setosa))[1,run(2,5)] contains row 1 (**nscores**) and columns 2 through 5 (**setosa**) of a 5 by 5 sample correlation matrix computed by **cor()**.

The correlation for variable 4 (W = .89172) is the smallest as we should have expected.

The critical values in the text don't apply exactly since they assume a slightly different definition of normal scores, but they should be very close.

The α = 1% value when n = 50 is .9671, so normality is rejected when W < .9671. This is the case only for variable 4.

But we are in a *multiple testing* situation. There are 4 ways to reject H_0: $\mathbf{X}$ is $N_4(\boldsymbol{\mu},\boldsymbol{\Sigma})$, so, when $\mathbf{X}$ is $N_4(\boldsymbol{\mu},\boldsymbol{\Sigma})$. there are 4 chances to make a type I error.

This means that the *actual* significance level

$$\alpha = \text{P(Reject } H_0 \text{ when it is true)}$$

is larger than α' = .01, the significance level used for each individual test.

Define the overall significance level α as

$$\alpha = \text{P(reject at least 1 } H_0 \,|\, \text{all } H_0 \text{ true)}$$

Then the **Bonferroni inequality** tells us that, when there are K tests (K = 4 here), each with significance level α', then α satisfies

$$\alpha' \leq \alpha \leq K\alpha'.$$

When α' is small, α is often quite close to $K\alpha'$.

This suggests you *Bonferronize* the tests.

You can do this in two ways:

- Use a modified critical value (cut point) which corresponds to significance level $\alpha' = \alpha/K$, where K is the number of tests. With K = 4, the text tables (with α' = .10, .05 and .01) allow only α = .40 = 4×.10, .20 = 4×.05 and .04 = 4×.01.
- Find modified P-values by multiplying the usual P-values by K and compare compare the Bonferronized P-values to the desired significance level α. The text tables don't allow for P-values at all.

How can you get Bonferronized P-values and/or critical values?

Often the easiest answer is **simulation**, generating many random samples for which H_0 is true and computing the test statistic from each of them.

Simulation approach

- Generate a large number M of N(0,1) samples for which you know H_0 is true.
- Compute W for each sample thus obtaining a random sample of size M from the null distribution of W
- From these M values, estimate P-values or critical values

```
Cmd> M <- 5000 # number of repetitions
Cmd> n <- nrows(setosa) # number of cases
Cmd> W <- rep(0,M) # room for the statistics
Cmd> for(i,1,M){
  W[i] <- cor(nscores,sort(rnorm(n)))[1,2] # 1,2 element of 2x2
;;}
```

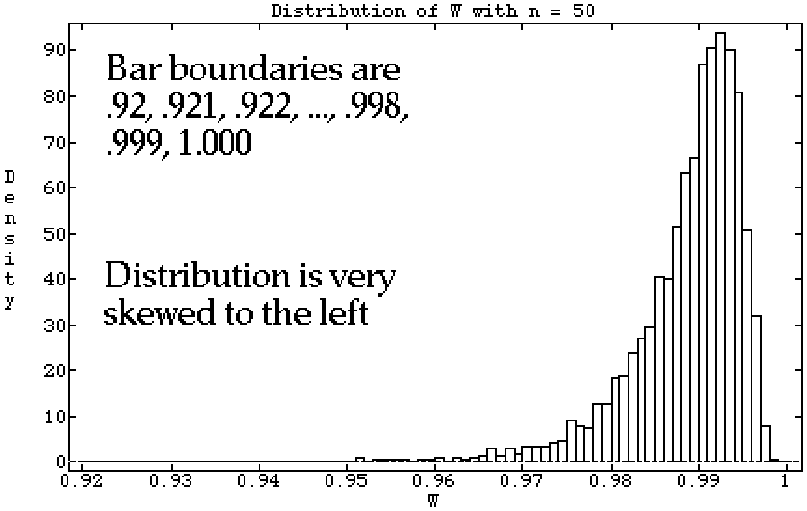
Each time through the loop, you

- Draw a standard normal random sample using `rnorm(n)` and order it by `sort()`
- Compute the correlation of the sorted values with the scores in `nscores`
- Stash the result in `W[i]`.

When it is done, `W` contains 5000 values of W computed when H_0 is true.

Here is what the sampling distribution looks like.

```
Cmd> hist(W,run(.92,1,.001),\
             title:"Distribution of W with n = 50")
```



Bar boundaries are .92, .921, .922, ..., .998, .999, 1.000

Distribution is very skewed to the left

Estimate P-values as sample proportions.

```
Cmd> sum(W < w') # counts of values in lower tail < w'
(1,1)        2685        613        164          0

Cmd> pvals <- sum(R < w')/M; pvals # approximate P-values
(1,1)       0.537      0.1226      0.0328         0
```

Since the W values follow the null distribution, the values in **pvals** are estimates of the actual P-values $P(W \leq W_{observed})$.

You estimate Bonferronized P-values by multiplying **pvals** by K = 4.

```
Cmd> K <- length(w) # number of tests

Cmd> K*pvals # 4*pvals = Bonferronized P-values
(1,1)       2.148      0.4904      0.1312          0
```

Only variable 4 as a really small P-value.

You can also estimate critical values and *Bonferronized* critical values as sample quantiles of the $\{W_i\}$.

```
Cmd> W <- sort(W) # 5000 ordered values

Cmd> J <- vector(.1,.05,.01)*(M+1)

Cmd> J # approximate indices of 10%, 5% and 1% quantiles
(1)        500.1       250.05        50.01

Cmd> floor(J) # round down (towards -oo)
(1)          500           250           50

Cmd> ceiling(J) # round up (towards +oo)
(1)          501           251           51

Cmd> .5*(W[floor(J)] + W[ceiling(J)]) # estimated quantiles
(1)    0.98066    0.97639    0.96635
```

These are non-Bonferronized critical values, quite close to the values ,9809, .9768, and .9671 in Table 4.2 in the text.

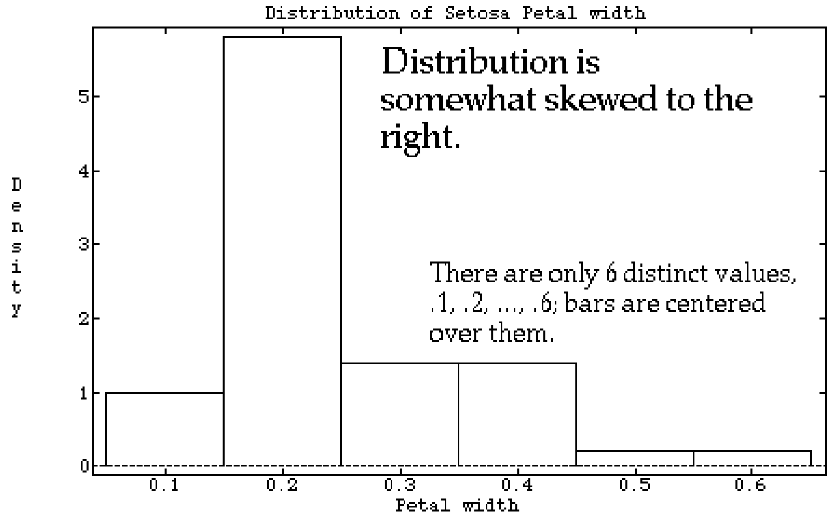
```
Cmd> J <- (vector(.1,.05,.01)/K)*(M+1) # K = 4

Cmd> .5*(W[floor(J)] + W[ceiling(J)]) # Bonferronized quantiles
(1)    0.97197      0.967    0.95316
```

These are <u>Bonferronized</u> critical values, that is critical values for α = .1/4, .05/4 and .01/4.

From either the P-values or critical value we see that only for X_4 is there strong evidence against normality.

Since at least one X_i appears to be non-normal, you can reject multivariate normality of **x**.

```
Cmd> hist(setosa,vector(.05,.1),\
title:"Distribution of Setosa Petal width",xlab:"Petal width")
```



Distribution is somewhat skewed to the right.

There are only 6 distinct values, .1, .2, ..., .6; bars are centered over them.

Conclusions:

- There is strong evidence that x_4 is not univariate normal. Hence the Setosa data isn't Multivariate normal.
- There is no significant evidence x_1, x_2 or x_3 are not univariate normal.

MacAnova note

`floor(x)` finds largest integer ≤ x (rounds up toward +∞)

`ceiling(x)` finds smallest integer ≥ x (rounds down toward -∞)

`round(x)` finds integer nearest to x

```
Cmd> floor(vector(-3.2,4.25,8))
(1)             -4            4            8

Cmd> ceiling(vector(-3.2,4.25,8))
(1)             -3            5            8

Cmd> round(vector(-3.2,4.25,8))
(1)             -3            4            8
```

A multivariate version

Let

$$d_j^2 \equiv (\mathbf{X}_j - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{X}_j - \boldsymbol{\mu}),\ j = 1,\ldots,n,$$

be the squared *Mahalanobis distances* of the data points from μ.

Then $\{d_1^2, d_2^2, \ldots, d_n^2\}$ constitute a *random sample* because they are

- independent
- have the same distribution.

When $\mathbf{X}$ is $N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$

- $d_1^2, d_2^2, \ldots, d_n^2$ are a random sample from χ_p^2.

If $\{d_i^2\}$ don't look like such a sample, H_0 may not be true.

In practice, since you don't know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, you estimate them by $\overline{\mathbf{X}}$ and $\mathbf{S}$, and calculate estimated values of d_i^2:

$$\hat{d}_j^2 = (\mathbf{X} - \overline{\mathbf{X}})'\mathbf{S}^{-1}(\mathbf{X} - \overline{\mathbf{X}})$$

At least in large samples you can treat $\hat{d}^2$ as if it were d^2.

This is not exact since

- $\{\hat{d}_1^2, \hat{d}_2^2, \ldots, \hat{d}_n^2\}$ is not a random sample (they are not independent)
- the distribution is not exactly χ_p^2 even when $\mathbf{X}$ is N_p but it's close enough.

MacAnova: Compute distances by

```
Cmd> d <- distcomp(x)
```

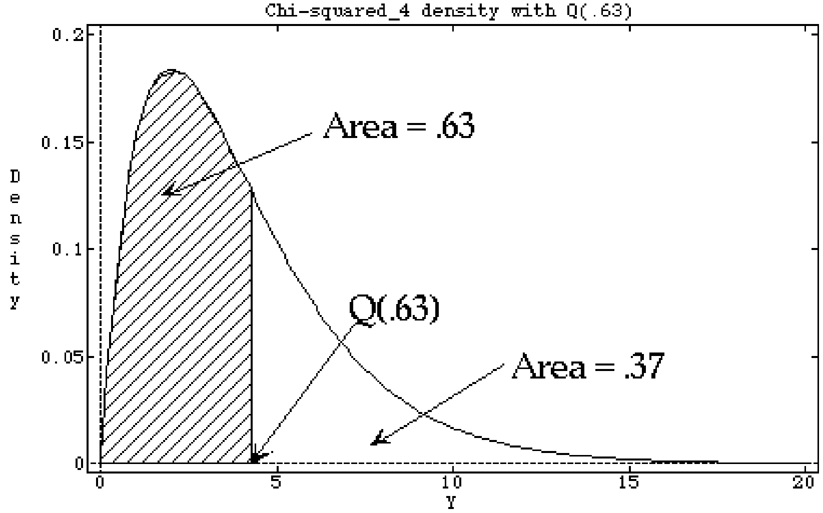
A ***Q-Q plot*** is a way to use a random sample to assess whether a random variable has a given distribution.

General case

Suppose Y_1, ..., Y_n is a univariate random sample from a random variable Y.

Let $F(y) = P(Y \leq y)$ be a *supposed* cumulative distribution function (CDF) for Y.

Let $Q(p) = F^{-1}(p)$, $0 \leq p \leq 1$, be the *supposed* p^{th} probability point of Y, that Q(p) satisfies $P(Y \leq Q(p)) = p$.



In MacAnova, you compute values of Q(p) for various distributions using **invnor()**, **invchi()**, **invF()**, etc.

A **Q-Q plot** is a scatter plot of

- *order statistics* $y_{(1)} \leq y_{(2)} \leq . . . \leq y_{(n)}$

against

- *probability points* $Q(p_1)$, ..., $Q(p_n)$, where $p_1 < p_2 < ... < p_n$ are *equally spaced* probabilities usually of the form $p_j = (j+\beta/2-.5)/(n+\beta)$, some β.

The most common choices for β are

β	p_j	Spacing
0	(j - .5)/n	1/n
.25	(j-3/8)/(n+1/4)	1/(n+1/4)
1	j/(n+1)	1/(n+1)

β = .25 is specifically recommended for the normal distribution and is what function **rankits()** uses for normal scores.

When F(y) actually *is* the CDF of Y, the plot should be approximately linear with slope 1 and intercept 0.

If it's sufficiently curved, that is evidence that F(y) is not the CDF of Y.

More generally, when the distribution of (Y - a)/c is F for some constants a and c, the Q-Q plot should be approximately linear with slope c and intercept a.

A *normal scores* plot is a Q-Q plot where F(x) = Φ(x) is the standard normal distribution.

Note that by definition, in a QQ plot, the points are always increasing (more precisely, never decreasing). This means the rank correlation will be 1 and the ordinary correlation will be high.

A χ^2 *Q-Q plot* is a useful way informally to *assess* whether d^2 is distributed as χ_p^2. As with a normal Q-Q plot, systematic curvature of the plotted points suggests the χ^2 distribution may not be appropriate.

A χ^2 Q-Q plot consists of two steps:

1. Order the calculated $\hat{d}_j^2$'s in increasing order (get order statistics)

$$\hat{d}_{(1)}^2 < \hat{d}_{(2)}^2 < \ldots < \hat{d}_{(n)}^2$$

MacAnova

If the $\hat{d}_i^2$'s are in vector **d**, you order them by **sort(d)**.

2. Plot the $\hat{d}_{(j)}^{2}$'s against chi-squared probability points computed using **invchi(q)**,

$$\chi_p^2(q_j),\ j = 1, 2, \ldots, n,$$

where $q_j = (j-.5)/n$, $j = 1, 2, \ldots, n$.

That is, $q_1 = (1/2)/n$, $q_2 = (3/2)/n$, $q_3 = (5/2)/n$, ..., $q_n = (n-1/2)/n$, are *equally spaced* probabilities. These satisfy $P(\chi_p^2 \leq \chi_p^2(q_j)) = q_j$

MacAnova

Compute the q_j by

```
  Cmd> q <- invchi((run(n)-.5)/n,p)
```

where p is the dimension (number of variables).

A Q-Q plot always *increases to the right*.

If d^2 is in fact χ_p^2, the plot should be approximately a *straight line through the origin* (0,0) with slope 1.

It is usually easier to assess a plot of

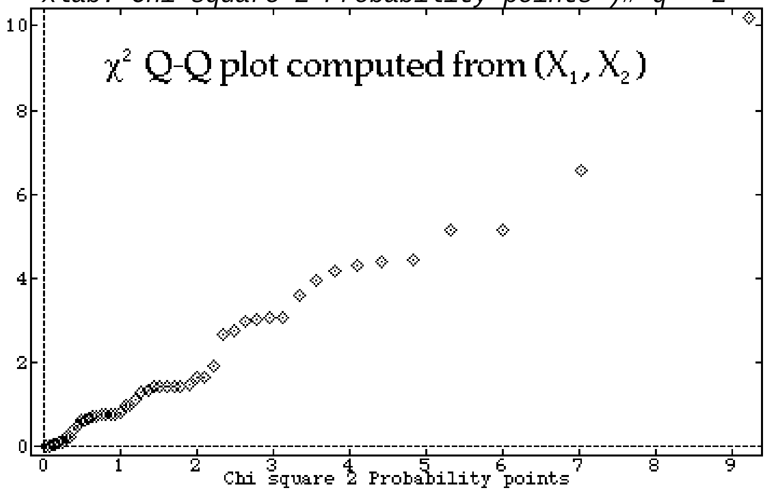
$$d_{(j)} = \sqrt{\{d_{(j)}^2\}} \text{ against } \sqrt{\{\chi_p^2(q_j)\}}$$

This should also be a straight line through the origin (0,0) when the data are normal

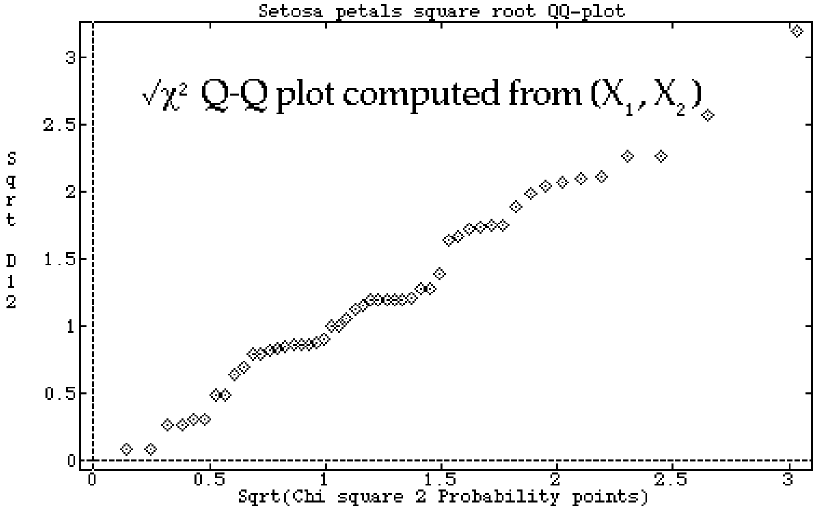
Note: Always *include the origin* (0, 0) in the plot. You do this in MacAnova by including **xmin:0,ymin:0** as arguments to the plotting command.

Do that with the Iris setosa data:

```
Cmd> n <- nrows(setosa)
Cmd> d12 <- distcomp(setosa[,run(2)])
Cmd> q2 <- invchi((run(n)-.5)/n,2) # d.f. = 2
Cmd> plot(q2, sort(d12),symbols:"\1",xmin:0,ymin:0,\
       title:"Setosa Petals QQ-plot", ylab:"D12",\
       xlab:"Chi square 2 Probability points")# q = 2
```



```
Cmd> plot(sqrt(q2),sqrt(sort(d12)),symbols:"\1",xmin:0,\
     ymin:0,xlab:"Sqrt(Chi square 2 Probability points)",\
     ylab:"Sqrt D12 ",title:"Setosa petals square root QQ-plot")
```



MacAnova Plotting Codes

There are several types and size of plotting codes you can use in graphs. You can get information on them by typing

```
Cmd> help(chplot:"drawn_plotting_symbols")
```

`plot(x,y,symbols:"\1")` uses large diamonds

`plot(x,y,symbols:"\14")` uses medium sized ×'s.

`plot(x,y,symbols:"\22")` uses small squares.

`plot(x,y,symbols:"\7")` uses dots visible by `addlines()`.