R blogs / lang
Articles about R, in your own language

Como fazer e interpretar o gráfico quantil-quantil

by Walmes Zeviani • November 30, 2012

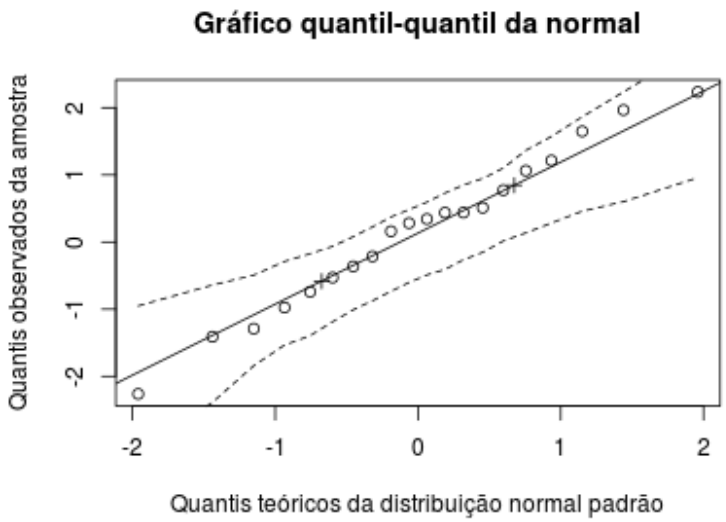


Gráfico quantil-quatil da distribuição normal com envelopes de confiança simulados.

O gráfico quantil-quantil (q-q) é uma ferramenta muito útil para checar adequação de distribuição de frequência dos dados à uma distribuição de probabilidades. Situações como essa ocorrem principalmente na análise de resíduos de modelos de regressão onde o gráfico q-q é usado para verificar se os resíduos apresentam distribuição normal. O gráfico q-q é melhor que o histograma e o gráfico de distribuição acumulada empírica porque nós temos mais habilidade para verificar se uma reta se ajusta aos pontos do que se uma curva de densidade se ajusta a um histograma ou uma curva de probabilidade acumulada se ajusta à acumulada empírica. Compare às três visualizações com o código a seguir.

```
#--------------------------------------------------------------------------
# q-q vs histograma vs ecdf

y <- rnorm(50)
par(mfrow=c(1,3))
qqnorm(y); qqline(y)
plot(density(y))
curve(dnorm(x, mean(y), sd(y)), add=TRUE, col=2)
plot(ecdf(y))
curve(pnorm(x, mean(y), sd(y)), add=TRUE, col=2)
```

```
11
12  # qual você sente mais segurança para verificar adequação?
13  # nossos olhos têm mais habilidade para comparar alinhamentos
14  #------------------------------------------------------------------------
```

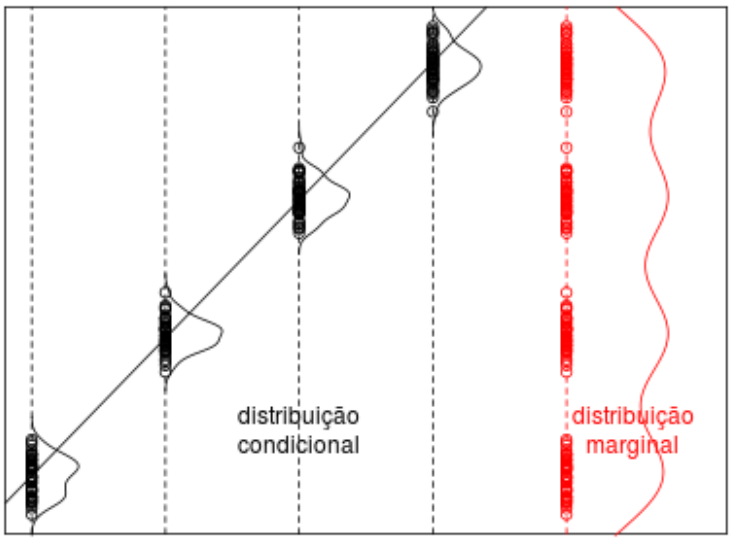
Apesar de muito usado, poucos usuários conhecem o procedimento para fazê-lo e interpretá-lo. O procedimento é simples e pode ser estendido para outras distribuições de probabilidade, não apenas para a distribuição normal como muitos podem pensar. Além do mais, alguns padrões do gráfico q-q obdecem à certas características dados dados, como assimetria, curtose e discreticidade. Saber identificar essas características é fundamental para indicar uma transformação aos dados. Abaixo o código para o gráfico q-q para distribuição normal, q-q para qualquer distribuição e o q-q com envelope de confiança obtido por simulação. A execução e estudo do código esclarece o procedimento. O envelope de confiança por simulação torna-se proibitivo para grandes amostras pelo tempo gasto na simulação.

```
1   #------------------------------------------------------------------------
2   # função para fazer o gráfico quantil-quantil da normal
3
4   qqn <- function(x, ref.line=TRUE){
5     x <- na.omit(x)               # remove NA
6     xo <- sort(x)                 # ordena a amostra
7     n <- length(x)                # número de elementos
8     i <- seq_along(x)             # índices posicionais
9     pteo <- (i-0.5)/n             # probabilidades teóricas
10    qteo <- qnorm(pteo)           # quantis teóricos sob a normal padrão
11    plot(xo~qteo)                 # quantis observados ~ quantis teóricos
12    if(ref.line){
13      qrto <- quantile(x, c(1,3)/4) # 1º e 3º quartis observados
14      qrtt <- qnorm(c(1,3)/4)       # 1º e 3º quartis teóricos
15      points(qrtt, qrto, pch=3)     # quartis, passa uma reta de referência
16      b <- diff(qrto)/diff(qrtt)    # coeficiente de inclinação da reta
17      a <- b*(0-qrtt[1])+qrto[1]    # intercepto da reta
18      abline(a=a, b=b)              # reta de referência
19    }
20  }
21
22  x <- rnorm(20)
23  par(mfrow=c(1,2))
24  qqn(x)
25  qqnorm(x); qqline(x)
26  layout(1)
27
28  #------------------------------------------------------------------------
29  # função para fazer o gráfico quantil-quantil de qualquer distribuição
30
31  qqq <- function(x, ref.line=TRUE, distr=qnorm, param=list(mean=0, sd=1)){
32    x <- na.omit(x)               # remove NA
33    xo <- sort(x)                 # ordena a amostra
34    n <- length(x)                # número de elementos
35    i <- seq_along(x)             # índices posicionais
36    pteo <- (i-0.5)/n             # probabilidades teóricas
37    qteo <- do.call(distr,        # quantis teóricos sob a distribuição
38                    c(list(p=pteo), param))
39    plot(xo~qteo)                 # quantis observados ~ quantis teóricos
40    if(ref.line){
41      qrto <- quantile(x, c(1,3)/4) # 1º e 3º quartis observados
42      qrtt <- do.call(distr,        # 1º e 3º quartis teóricos
43                      c(list(p=c(1,3)/4), param))
44      points(qrtt, qrto, pch=3)     # quartis, por eles passa uma reta de referênc
45      b <- diff(qrto)/diff(qrtt)    # coeficiente de inclinação da reta
46      a <- b*(0-qrtt[1])+qrto[1]    # intercepto da reta
47      abline(a=a, b=b)              # reta de referência
48    }
49  }
50
51  x <- rnorm(20)
```

```r
par(mfrow=c(1,2))
qqq(x)
qqnorm(x); qqline(x)
layout(1)

x <- runif(20)
qqq(x, ref.line=TRUE, distr=qunif, param=list(min=0, max=1))

x <- rgamma(20, shape=4, rate=1/2)
qqq(x, ref.line=TRUE, distr=qgamma, param=list(shape=4, rate=1/2))

#-------------------------------------------------------------------------
# envelope para o gráfico de quantis (simulated bands)

qqqsb <- function(x, ref.line=TRUE, distr=qnorm, param=list(mean=0, sd=1),
                  sb=TRUE, nsim=500, alpha=0.95, ...){
  x <- na.omit(x)               # remove NA
  xo <- sort(x)                 # ordena a amostra
  n <- length(x)                # número de elementos
  i <- seq_along(x)             # índices posicionais
  pteo <- (i-0.5)/n             # probabilidades teóricas
  qteo <- do.call(distr,        # quantis teóricos sob a distribuição
                  c(list(p=pteo), param))
  plot(xo~qteo, ...)            # quantis observados ~ quantis teóricos
  if(ref.line){
    qrto <- quantile(x, c(1,3)/4) # 1º e 3º quartis observados
    qrtt <- do.call(distr,        # 1º e 3º quartis teóricos
                    c(list(p=c(1,3)/4), param))
    points(qrtt, qrto, pch=3)     # quartis, passa uma reta de referência
    b <- diff(qrto)/diff(qrtt)    # coeficiente de inclinação da reta
    a <- b*(0-qrtt[1])+qrto[1]    # intercepto da reta
    abline(a=a, b=b)              # reta de referência
  }
  if(sb){
    rdistr <- sub("q", "r",       # função que gera números aleatórios
                  deparse(substitute(distr)))
    aa <- replicate(nsim,         # amostra da distribuição de referência
                    sort(do.call(rdistr, c(list(n=n), param))))
    lim <- apply(aa, 1,           # limites das bandas 100*alpha%
                 quantile, probs=c((1-alpha)/2,(alpha+1)/2))
    matlines(qteo, t(lim),        # coloca as bandas do envelope simulado
             lty=2, col=1)
  }
}

x <- rnorm(20)

#png("f047.png", 400, 300)
qqqsb(x, xlab="Quantis teóricos da distribuição normal padrão",
      ylab="Quantis observados da amostra",
      main="Gráfico quantil-quantil da normal")
#dev.off()

x <- rpois(50, lambda=20)
qqqsb(x, distr=qpois, param=list(lambda=20))

#-------------------------------------------------------------------------
```

A normalidade dos resíduos é um dos pressupostos da análise de regressão. **São os resíduos e não os dados que devem apresentar normalidade**. Se a distribuição dos dados, ou melhor, da sua variável resposta (Y) condicional ao efeito das suas variáveis explicativas for normal, os resíduos terão distribuição normal. Porém, se você aplica um teste de normalidade aos dados (Y) você não está considerando os efeitos das variáveis explicativas, ou seja, você está aplicando um teste na distribuição marginal de Y que não tem porque atender a normalidade. Todo teste de normalidade supõe que os dados têm uma média e

uma variância e só os resíduos atendem essa premissa porque os dados (Y) têm média dependente do efeito das variáveis explicativas.



Distribuição condicional e marginal da variável resposta.

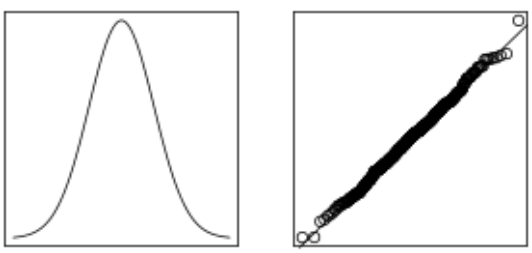
```
#------------------------------------------------------------------------------
# distribuição condicional vs distribuição marginal

layout(1)
x <- rep(1:4, e=50)
y <- rnorm(x, mean=0+0.75*x, sd=0.1)
da <- data.frame(x, y)
plot(y~x, da)

db <- split(da, f=da$x)

#png("f046.png", 400, 300); par(mar=c(1,1,1,1))
plot(y~x, da, xlim=c(1,6), xaxt="n", yaxt="n", xlab="", ylab="")
abline(a=0, b=0.75)
lapply(db,
       function(d){
         dnst <- density(d$y)
         lines(d$x[1]+dnst$y*0.1, dnst$x)
         abline(v=d$x[1], lty=2)
       })
points(rep(5, length(y)), da$y, col=2)
abline(v=5, lty=2, col=2)
dnst <- density(da$y)
lines(5+dnst$y*2, dnst$x, col=2)
text(3, 1, "distribuição\ncondicional")
text(5.5, 1, "distribuição\nmarginal", col=2)
#dev.off()

par(mfrow=c(1,2))
qqnorm(da$y)
qqnorm(residuals(lm(y~x, da)))
layout(1)

#------------------------------------------------------------------------------
```

Muitos usuários preferem aplicar um teste de normalidade do que olhar para o gráfico q-q. Isso têm duas razões: (1) costume, o usuário sempre usou aplicativos para análise de dados que não dispõem de recursos gráficos, eles conduzem toda análise sem sequer ver os dados em um gráfico, (2) consideram subjetiva a análise gráfica. Do meu ponto de vista, a subjetividade está presente ao aplicar um teste também pois o

usuário é quem escolhe o teste e o nível de significância. Mas o que de fato eu defendo é que a análise gráfica é indiscutivelmente mais informativa. Veja, se o teste rejeita a normalidade é porque os dados não apresentam distribuição normal por algum motivo. Quando você visualiza o q-q é possível explicar a fuga de normalidade que pode ser sistemática: (1) devido à desvio de assimetria, (2) de curtose, (3) à mistura de distribuições, (4) à presença de um outlier e (5) ao dados serem discretos. Essas são as principais causas de afastamento. Cada uma delas sugere uma alternativa para corrigir a fuga: tranformação, modelagem da variância, deleção de outlier, etc. Nesse sentido, como identificar esses padrões de fuga? É o que os gráficos animados vão mostrar. Até a próxima ridícula.

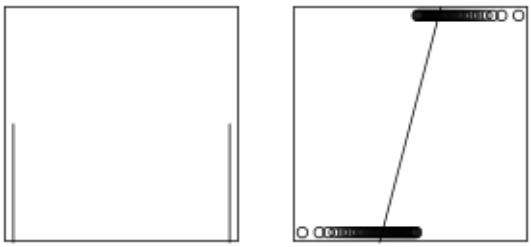
Fuga de normalidade por assimetria representada pelo gráfico quantil-quantil. A assimetria no q-q aparece com pontos dispostos em forma de arco.

```
#------------------------------------------------------------------------
# assimetria

dir.create("frames")
setwd("frames")

png(file="assimet%03d.png", width=300, height=150)
par(mar=c(1,1,1,1))
par(mfrow=c(1,2))
for(i in 10*sin(seq(0.01, pi-0.01, l=100))){
  curve(dbeta(x, i, 10-i), 0, 1, xaxt="n", yaxt="n", xlab="", ylab="")
  y <- rbeta(100, i, 10-i)
  qqnorm(y, xaxt="n", yaxt="n", main=NULL, xlab="", ylab=""); qqline(y)
}
dev.off()

# converte os pngs para um gif usando ImageMagick
system("convert -delay 10 assimet*.png assimet.gif")

# remove os arquivos png
file.remove(list.files(pattern=".png"))

#------------------------------------------------------------------------
```



Fuga de normalidade por mistura de distribuições (alteração de curtose) representada pelo gráfico quantil-quantil. Mistura de distribuições geram fugas nos extremos.

```r
#-------------------------------------------------------------------------
# mistura de distribuições

png(file="mistura%03d.png", width=300, height=150)
par(mar=c(1,1,1,1))
par(mfrow=c(1,2))
for(i in sin(seq(0, pi, l=100))){
  curve(i*dnorm(x,0,1)+(1-i)*dnorm(x,0,6), -20, 20,
        xaxt="n", yaxt="n", xlab="", ylab="")
  y <- c(rnorm(ceiling(500*i),0,1), rnorm(floor(500*(1-i)),0,6))
  qqnorm(y, xaxt="n", yaxt="n", main=NULL, xlab="", ylab=""); qqline(y)
}
dev.off()

# converte os pngs para um gif usando ImageMagick
system("convert -delay 10 mistura*.png mistura.gif")

# remove os arquivos png
file.remove(list.files(pattern=".png"))

#-------------------------------------------------------------------------
```



Fuga de normalidade devido dados serem discretos (padrão escada) representada pelo gráfico quantil-quantil.

```r
#-------------------------------------------------------------------------
# discreticidade

png(file="discret%03d.png", width=300, height=150)
par(mar=c(1,1,1,1))
par(mfrow=c(1,2))
for(i in c(1:100, 99:1)){
  x <- 0:i
  px <- dbinom(x, i, 0.5)
  plot(px~x, type="h", xaxt="n", yaxt="n", xlab="", ylab="")
  y <- rbinom(100, i, 0.5)
  qqnorm(y, xaxt="n", yaxt="n", main=NULL, xlab="", ylab=""); qqline(y)
}
dev.off()

# converte os pngs para um gif usando ImageMagick
system("convert -delay 10 discret*.png discret.gif")

# remove os arquivos png
file.remove(list.files(pattern=".png"))

#-------------------------------------------------------------------------
```

Tags: animação gif inferência normalidade quantil

- Ile punktów potrzeba by się dostać do szkoły średniej w Warszawie?

← 决策树之三国争霸

Eine kurze Geschichte über R →

LANGUAGES

- Chinese
- Dutch
- French
- German
- Indonesian
- Italian
- Korean
- Polish
- portuguese
- Russian
- Serbian
- Spanish
- Uncategorized

RECENT POSTS

- Tell Me a Story: How to Generate Textual Explanations for Predictive Models
- dime: Deep Interactive Model Explanations
- Learn about XAI in R with ,,Predictive Models: Explore, Explain, and Debug”
- Communautés de prénoms