

MINI CURSO

ESTATÍSTICA APLICADA A MELHORIA DE PROCESSOS COM A LINGUAGEM DE PROGRAMAÇÃO R

GUIA DE USUÁRIO

SUMÁRIO

PARA NOSSOS LEITORES	3
Pensando no futuro	3
LINGUAGEM E AMBIENTE R	4
Instalando o R	4
Conhecendo o R-Studio	5
Instalando e ativando as bibliotecas	6
Conceitos básicos da linguagem de programação R	7
ESTATÍSTICA DESCRITIVA EM R	9
Medidas de resumo	9
Medidas de dispersão	10
Distribuição normal	10
Plotando gráficos estatísticos no R	11
CONTROLE ESTATÍSTICO DE PROCESSOS (CEP)	16
Cartas de controle	16
Cartas de controle para variáveis	16
Cartas de controle para atributos	18
Capabilidade do processo	19
índice de capacidade (C_p)	20
Índice de desempenho do processo (C_{pK})	21
REFERÊNCIA	23

PARA NOSSOS LEITORES

“Melhoria de Processos é uma iniciativa específica ou um projeto para melhorar o alinhamento e o desempenho de processos com a estratégia organizacional e as expectativas do cliente”

Vinicius Nóbile de Almeida

Pensando no futuro

A melhoria de processos surge com o intuito de nivelar os mecanismos dentro da organização de maneira que todos funcionem de forma controlada do início da cadeia produtiva ao fim. Nas indústrias, a melhoria de processos assumem o papel de identificador de gargalos, falhas e deformidades, tornando os processos mais produtivos e eficientes. Isso, garante-se que tais, supram às exigências dos clientes e alcance os objetivos esperados.

Para tal, deve-se analisar o processo atual afim de compreender como ele pode ser melhorado e montar o fluxo de trabalho do processo para agregar valor ao cliente. Existem diversas formas de realizar a melhoria de processos, dependendo da abordagem escolhida, pode-se fazer o uso de ferramentas estatísticas.

É aí que podemos contar com o controle estatístico de processo (CEP), um método de gestão da qualidade que utiliza ferramentas e técnicas estatísticos para identificar potenciais falhas de um determinado processo. Com ele é possível reduzir ou, até mesmo, eliminar, possíveis causas de variações, que podem ser corrigidas ao observar o processo. Com a avaliação de indicadores, o CEP permite uma estabilidade no processo, garantindo previsibilidade aos gestores e eliminando riscos ou surpresas.

Mas por que usar a linguagem de programação estatística R?

O R oferece uma gama de soluções personalizadas, incluindo análise robusta de dados e gráficos dinâmicos. Além de funcional, a linguagem conta com diversas bibliotecas capazes de formular cartas de controle, bem como determinar os índices de capacidade dos processos e máquinas.

Equipe Mini Curso

Alexandre Marzochi

Gabriella Stella

Jeiciane S Paula

Victoria Schettini

LINGUAGEM E AMBIENTE R

A linguagem de programação estatística e ambiente R é considerada multi-paradigma, pois está voltada a programação funcional, de caráter dinâmico e fracamente tipada. Seu desenvolvimento teve origem no departamento de Estatística da Universidade de Auckland, por Ross Ihaka e Robert Gentleman, orientada à manipulação, análise e visualização de dados. Atualmente, é gerenciada por uma comunidade de colaboradores voluntários que cooperam com código fonte da linguagem e com a ampliação de funcionalidades por bibliotecas. A linguagem R é amplamente usada em Data Analytics, sua interface possibilita a utilização de suas funcionalidades de maneira acessível, prática e visual. Basicamente, a estrutura dos códigos dar-se por linhas de comando, das quais são executadas de forma sequenciada.

Instalando o R

Primeiramente é necessário baixar o software R e seu ambiente de desenvolvimento integrado, denominado R-Studio, o ambiente será utilizado para manipulação e execução dos códigos. Na página <https://cran.r-project.org/bin/windows/base/> é possível baixar o software R, ou acessar o instalador abaixo.

R-4.0.2 for Windows (32/64 bit)

[Download R 4.0.2 for Windows](#) (84 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN-MIRROR>/bin/windows/base/release.html](#).



Figura 1: Primeiro passo para baixar o R

Após a finalização do download e instalação do software R (4.0.2), abra a página oficial do R-Studio <https://rstudio.com/products/rstudio/download/>, e instale ambiente de desenvolvimento integrado. No final da página estará todos os instaladores disponíveis.

All Installers

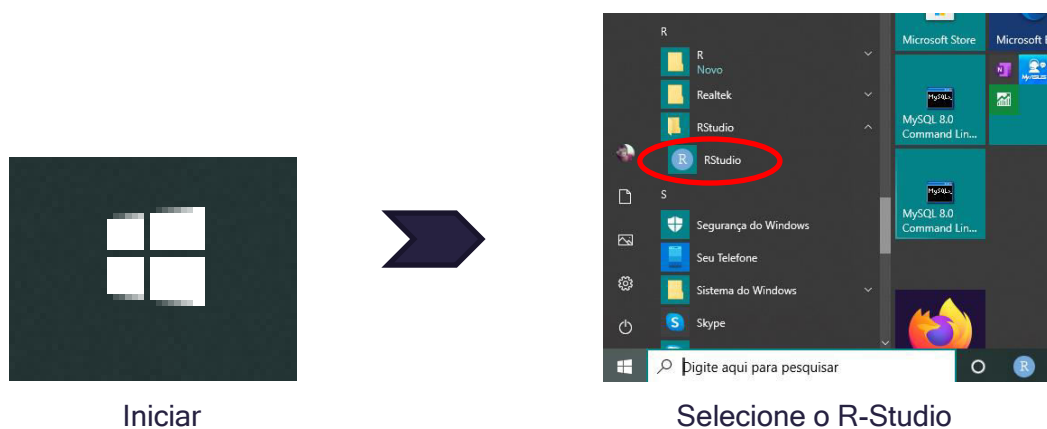
Linux users may need to [import RStudio's public code-signing key](#) prior to installation, depending on the operating system's security policy.

RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an [older version of RStudio](#).

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.3.1073.exe	171.62 MB	2fea472a
macOS 10.13+	RStudio-1.3.1073.dmg	148.66 MB	0878b395
Ubuntu 16	rstudio-1.3.1073-amd64.deb	124.07 MB	6d71c5ff
Ubuntu 18/Debian 10	rstudio-1.3.1073-amd64.deb	126.78 MB	86be9352
Fedora 19/Red Hat 7	rstudio-1.3.1073-x86_64.rpm	146.95 MB	01abb3d8
Fedora 28/Red Hat 8	rstudio-1.3.1073-x86_64.rpm	151.04 MB	4b4e4878
Debian 9	rstudio-1.3.1073-amd64.deb	126.98 MB	0226bbc2
SLES/OpenSUSE 12	rstudio-1.3.1073-x86_64.rpm	119.43 MB	7c1a6f2c
OpenSUSE 15	rstudio-1.3.1073-x86_64.rpm	128.39 MB	29078f11

Figura 2: Instalando o R-Studio

Abra o R-Studio para desenvolver e executar os códigos R. Para localizar o Software basta seguir os seguintes passos:



Conhecendo o R-Studio

Ao executar R-Studio, o ambiente de interação com o usuário será aberto. Através da figura 3, observa-se quatro quadrantes: Editor, console, environment e o output.

As principais funções dos painéis são:

1. **Editor/Scripts:** local onde é escrito as linhas de códigos.
2. **Console:** Local em que código são executados e recebimento das saídas.
3. **Environment:** Painel de visualização dos objetos criados durante a sessão.
4. **Files:** Expõe os arquivos presente no diretório de trabalho.
5. **Plots:** Painel onde os gráficos serão expostos.
6. **Help:** Janela de apresentação da documentação das funções.
7. **History:** Painel com um histórico dos comandos rodados.

Para criar um arquivo R .R basta ir em **File > New File > R Scripts**, para abrir um código R já elaborado, basta ir em **File > open File > localizar o diretório do arquivo > clicar em open**.

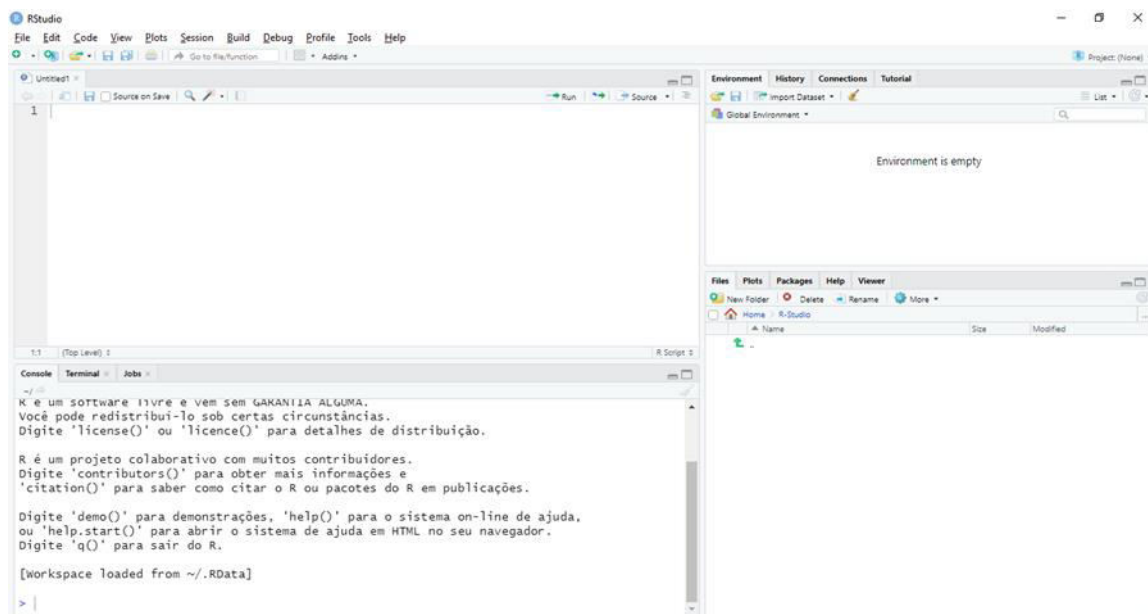


Figura 3: Ambiente R-Studio

Instalando e ativando as bibliotecas

No R, um grande volume de pacotes. Tais pacotes fornecem um conjunto de funções que viabilizam a realização das análises estatísticas, além de possuírem documentações que auxiliam no entendimento das funções e demonstrações de execução. Ao instalar o R, alguns pacotes padrões são baixados automaticamente, os quais são essenciais para o funcionamento do programa, e são conhecidos como pacote básico ou módulo.

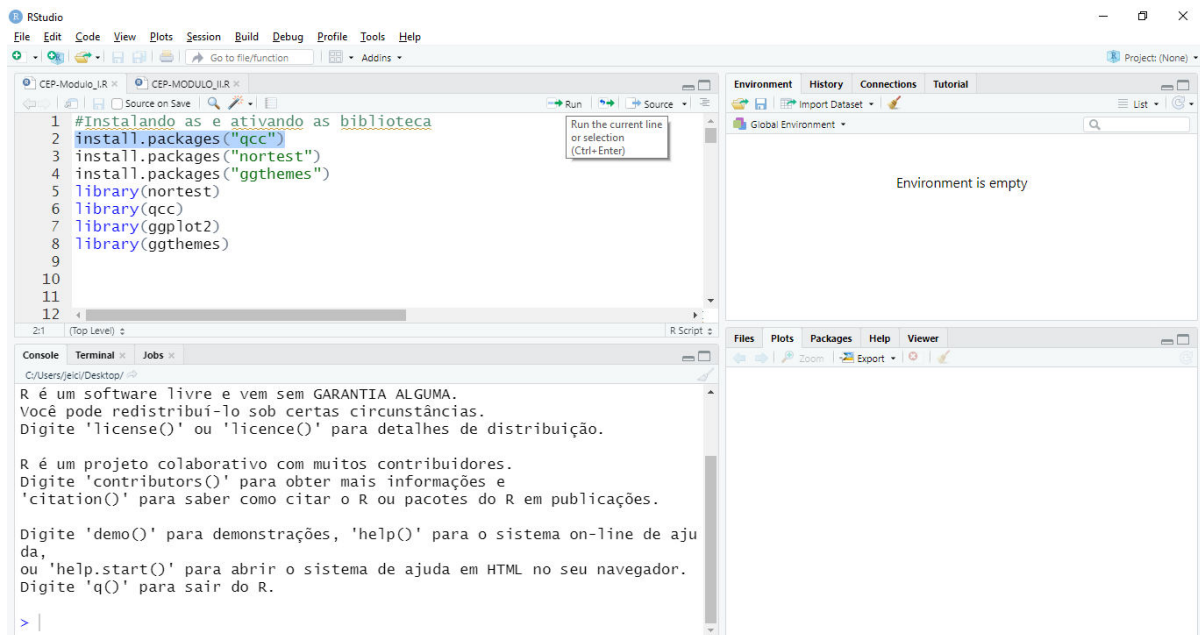
Existem duas maneiras de encontrar os pacotes extras, a primeira trata-se da utilização do próprio programa, a segunda é através do site www.r-project.org. Em ambos os casos, o usuário deverá estar conectado à Internet. Instalando pacotes a partir do R, deve-se seguir as instruções abaixo:

8. Clique em “pacotes” no menu;
9. Clique em “instalar pacote”;
10. Selecione o nome do pacote que deseja, clique em “OK”;
11. Espere o término do download ser concluído;

12. O console do R mostrará um texto indicando que o pacote escolhido foi instalado.

Outra opção, é escrever no Editor a linha de comando `install.packages("")`, e inserir dentro do colchete o nome da biblioteca a ser instalada. Este método foi escolhido para compor o manual devido sua acessibilidade.

Para ativar a biblioteca, basta utilizar a linha de comando `library()` como o nome da referida entre colchetes, porém sem as aspas.



Conceitos básicos da linguagem de programação R

De forma simplificada, o ambiente R foi desenvolvido voltado a realização de operações matemáticas complexas, assim, é atribuído, normalmente, estruturas de dados como vetores, matrizes, listas e arrays. Estes, podem ser classificados como:

- **Vetor**, trata-se de um conjunto de dados, que detém uma dimensão e é criado pela função `c()`, conhecida como concatenar. No R, não é possível criar vetores vazios.
- **Matriz**, é um conjunto de dados que possuem o mesmo tipo de dados e são organizados em duas dimensões (linha e colunas). Em R, usa-se a função `matrix()`, tendo os parâmetros `"ncol"` (número de colunas) e `"nrow"` (número de linhas).
- **Array**, é agrupamento multidimensional de dados, em que as linhas cotem o mesmo comprimento da quantidades de colunas, a função usada para criar um array é `array()`.
- **Lista**, refere-se a um objeto capaz de armazenar outros objetos de tipos e tamanhos diversos, logo, cada entidade que compõe a lista, pode ser de um tipo diferente.
- **Data.frame**, é uma estrutura de dados bidimensional, em que cada coluna pode enquadrar tipos de dados distinto. O `data.frame` é similar a uma tabela de dados, a função `data.frame()` descreve a estrutura.

Os códigos para algumas estruturas são descritos abaixo.

```
codigo = c(1,2,3)
nome= c("Jeiciane", "Samuel", "Victoria")
nome
```

```
## [1] "Jeiciane" "Samuel" "Victoria"

matriz= matrix(c(1,2,3,"Jeiciane", "Samual", "Victoria"), nrow = 3, ncol = 2)
matriz

##      [,1] [,2]
## [1,] "1"  "Jeiciane"
## [2,] "2"  "Samual"
## [3,] "3"  "Victoria"

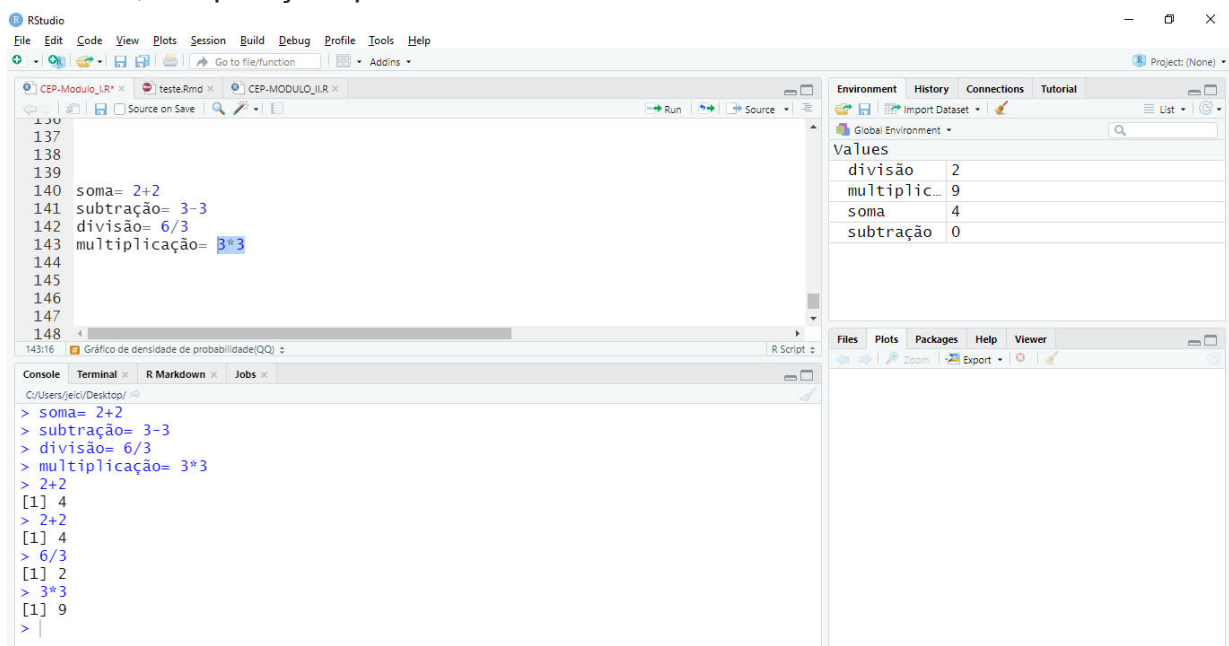
Data.frame = data.frame(codigo, nome)
Data.frame

##   codigo   nome
## 1      1 Jeiciane
## 2      2 Samuel
## 3      3 Victoria
```

O R, permite realizar algumas operações matemáticas, como soma, subtração, divisão, multiplicação etc. A tabela abaixo mostra alguns operadores utilizados em R.

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
^	Potenciação
=	Igual
!=	Diferente
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a
&	E lógico
	OU lógico
!	Não lógico

No R-Studio, as operações podem ser realizadas diretamente no console.



ESTATÍSTICA DESCRITIVA EM R

Ao analisar uma determinada característica relente de um produto ou serviço mediante um conjunto de medidas, espera-se definir um padrão de comportamento dos dados em sua totalidade. Tal padrão, caracteriza-se por ínfimos números e gráficos que dimensionam e fornecem a visualização de informações importantes.

A estatística descritiva, busca sumarizar as particularidades de um agrupamento de dados, por meio de técnicas analíticas. Assim, é possível definir, números que assinalam a posição central dos dados, como as medidas resumo (média, moda, mediana, quartis etc.), e as medidas de dispersão (desvio padrão, variância etc.).

Medidas de resumo

Por se tratar de um mini curso de Controle Estatístico de Processo, serão abordadas as medidas resumo ou de posição centrar, média e mediana, mais usuais no CEP.

A média aritmética é a soma das observações dividida pelo número de dados usados na análise, logo, dado um conjunto de dados x_1, x_2, \dots, x_n

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n}$$

No R, a função Rbase ***mean(x)*** calcula a média dos elementos da variável x.

```
#Media
mean(Dados$diameter)

## [1] 74.0036
```

A mediana é a medida que divide um certo conjunto de dados, ou seja, 50% dos valores encontram-se acima da mediana e a outra metade abaixo. Alguns procedimentos são adotados para encontrar a mediana de um conjunto de dados, primeiramente organize os dados em ordem crescente, caso a amostra tenha tamanho ímpar, a mediana é dada por $a[(n + 1)/2]$, ou seja, ésima posição dos dados organizados. Para tamanho amostra par, utiliza-se como mediana a média aritmética dos dois elementos centrais.

Em R a função Rbase ***median(x)*** define a mediana de uma amostra.

```
#Mediana
median(Dados$diameter)

## [1] 74.003
```

Medidas de dispersão

As medidas de dispersão, buscam sumarizar a variabilidade de um conjunto de dados, tais, permitem, por exemplo, conjuntos de valores distintos de acordo com algum critério estabelecido previamente.

Entre as medidas de dispersão, tem-se a variância, um valor pequeno indica que cada valor pertencente ao conjunto está próximo do valor médio central. A variância é calculada pela média dos quadrados das diferenças. Em R a função **var()** calcula a variância da amostra.

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}$$

```
#Medida de Posição e Dispersão
```

```
var(Dados$diameter)
```

```
## [1] 0.0001303507
```

O desvio padrão é a raiz quadrada da variância, e espelha a variação dos dados de um conjunto, induzindo a uniformidade dele. Assim, um desvio padrão pequeno, pode indicar homogeneidade nos dados. Em R a função **sd()**, calcula o desvio padrão da amostra.

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$$

```
#Desvio Padrão
```

```
sd(Dados$diameter)
```

```
## [1] 0.01141712
```

A função Rbase **summary()**, retorna o sumário dos dados com algumas medidas de tendência central, como, média, mediana, 1º quartil, 3º quartil, mínimo e máximo.

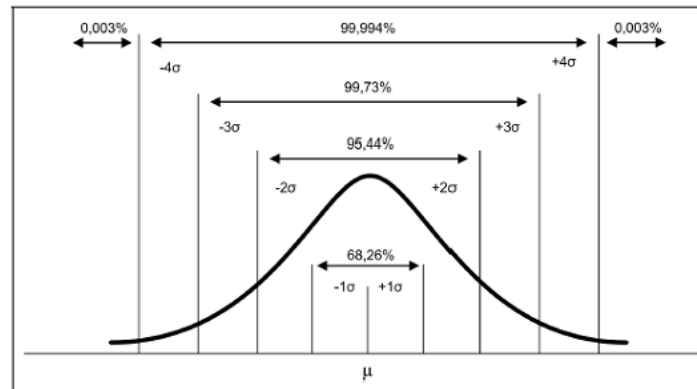
```
summary(Dados$diameter)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  73.97   74.00   74.00   74.00   74.01   74.04
```

Distribuição normal

A distribuição normal ou Gaussiana, remete a curva que em relação ao ponto médio possui simetria, visualmente similar a um sino. Os processos assumem formas variáveis de distribuição de frequências que em suma, são próximos a distribuição de probabilidade

normal. Define-se a probabilidade como a oportunidade de acontecer algum evento, ou seja, a possibilidade de ocorrer uma medida em um certo intervalo.



A curva normal é descrita por dois parâmetros estudados anteriormente, a média e o desvio padrão populacional. Na figura acima, estão representadas as áreas importantes da curva.

Na linguagem de programação estatística R, existem várias maneiras para saber se o conjunto de dados é normal. Dois testes de hipóteses amplamente utilizados são: Shapiro-Wilk e Anderson-Darling, em ambos testes averigua se uma determinada amostra aleatória segue uma distribuição normal. As hipóteses adotadas pelos testes são:

$$\begin{cases} H_0 = \text{As distribuições são normais} \\ H_1 = \text{As distribuições não são normais} \end{cases}$$

Para casos em que o valor de p é menor que o nível alfa (significância) escolhido, a hipótese nula é rejeitada evidenciando que dados testados não são normalmente distribuídos.

A função `shapiro.test()` é refere-se ao teste de Shapiro-Wilk e `ad.test()` ao Anderson-Darling.

```
#Inferência
Sh = shapiro.test(Dados$diameter) # Shapiro-Wilk
An = ad.test(Dados$diameter) # Anderson-Darling
print(Sh)

##
## Shapiro-Wilk normality test
##
## data: Dados$diameter
## W = 0.98968, p-value = 0.1607

print(An)

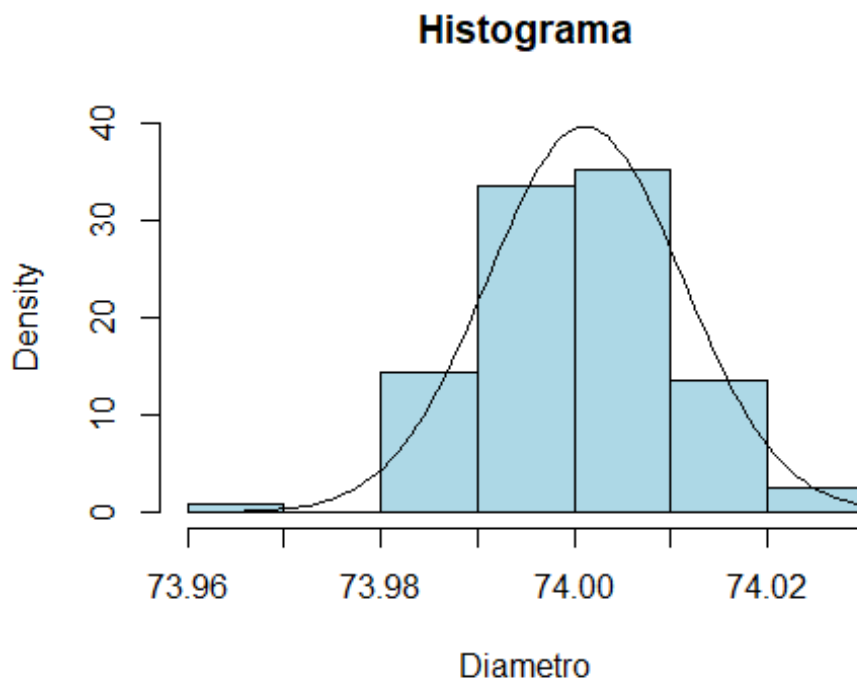
##
## Anderson-Darling normality test
##
## data: Dados$diameter
## A = 0.51807, p-value = 0.1862
```

Plotando gráficos estatísticos no R

A linguagem de programação R permite confeccionar os gráficos estáticos por dois caminhos distintos, no primeiro caso usa-se as funções da Rbase, a outra opção é através do pacote “ggplot2”. Neste manual, será abordado apenas o primeiro caminho para os gráficos mais comuns e que serão utilizados em seções posteriores como o histograma e boxplot. O código usando o “ggplot2”, está disponibilizado abaixo.¹

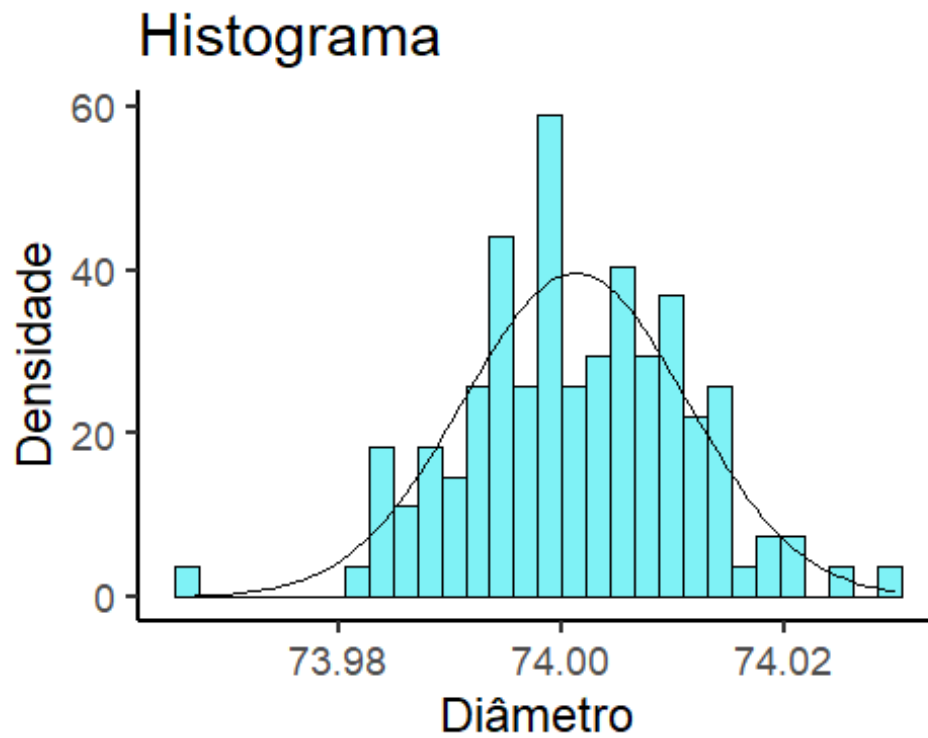
O gráfico de barras, histograma, espelha a distribuição da frequência de um determinado conjunto de dados contínuos e pode ser constituído por valores absolutos, relativos e densidades. A função *hist()* cria o histograma, os parâmetros mais comuns atribuídos são; “main”, “col”, “xlab” e “ylab”.

```
hist(fase1$diameter,
     main = "Histograma", #função base do R
     xlab = "Diametro",
     col = "lightblue",
     freq = F,
     breaks = 5, ylim = c(0,40))
curve(dnorm(x,mean = mean(fase1$diameter),
                  sd = sd(fase1$diameter)), add = T)
```



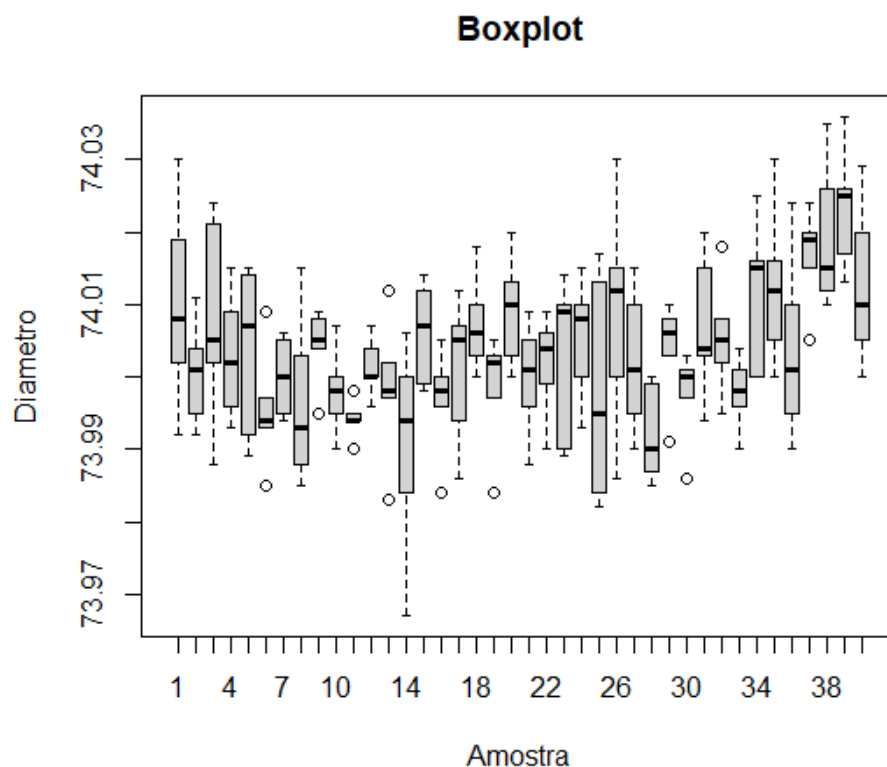
Usando o pacote ggplot2 para fazer o histograma com a curva de densidade.

```
ggplot(Dados, aes (x= diameter, fill = trial)) +
  geom_histogram(bins = 30, color = "white",
                 position = "dodge") +
  theme_classic(base_size = 18) + labs(title = "Histograma")+
  xlab("Diametro") + ylab("Frequência")
```

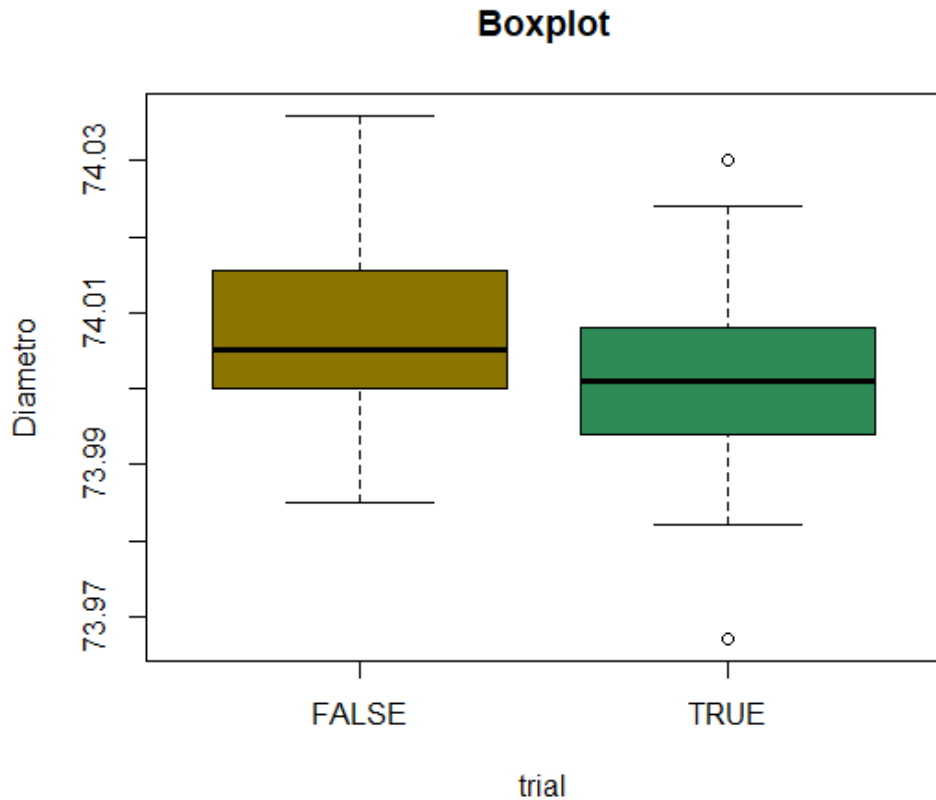


Conhecido também como diagrama de caixas, é uma ferramenta gráfica utilizada para visualizar a distribuição e valores discrepantes dos dados. A função usada pelo Rbase é *boxplot()*, e *ggplot()* em conjunto com o *geom_boxplot()* para a biblioteca ggplot2.

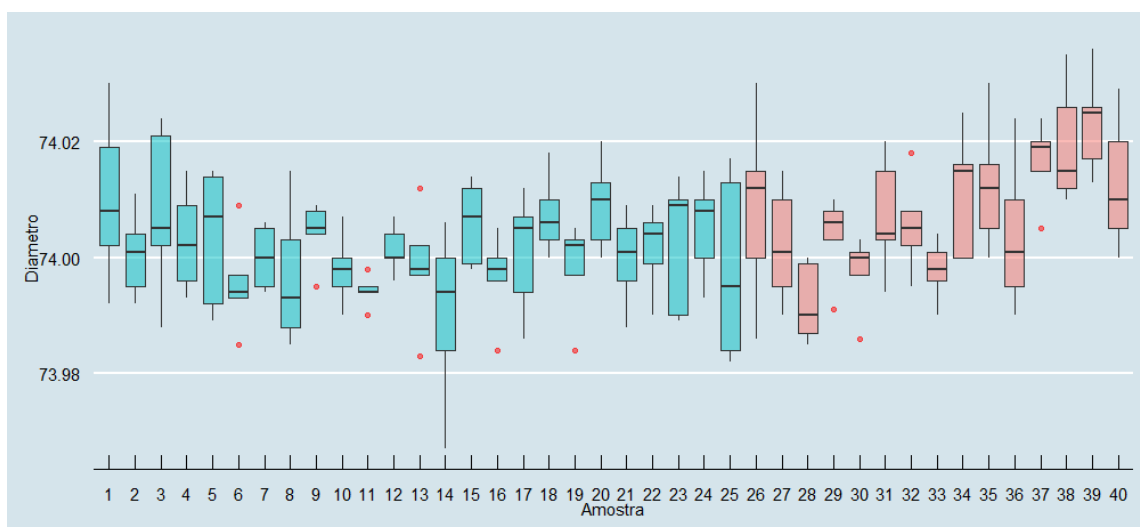
```
boxplot(diameter ~ sample,
  main = "Boxplot", #função base do R
  xlab = "Amostra",
  ylab = "Diametro",
  breaks = 5) #Por Amostra
```



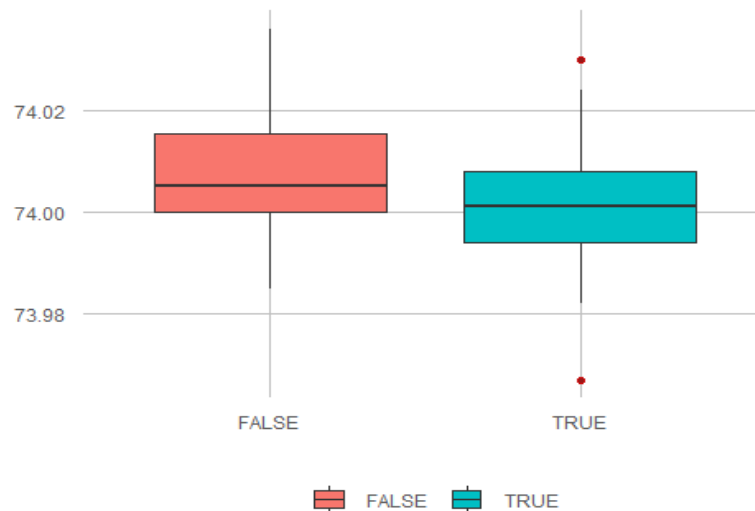
```
boxplot(diameter ~ trial,
  main = "Boxplot", #função base do R
  xlab = "Amostra",
  ylab = "Diametro",
  breaks = 5,
  col= c("gold4"," seagreen4")) #Por fase
```



```
ggplot(Dados, aes (y=diameter, x= factor(sample),fill = trial))+
  xlab("Amostra") + ylab("Diametro")+
  geom_boxplot(show.legend = F, alpha = 0.5, outlier.color = "red")+
  theme_economist(base_size = 18)
```

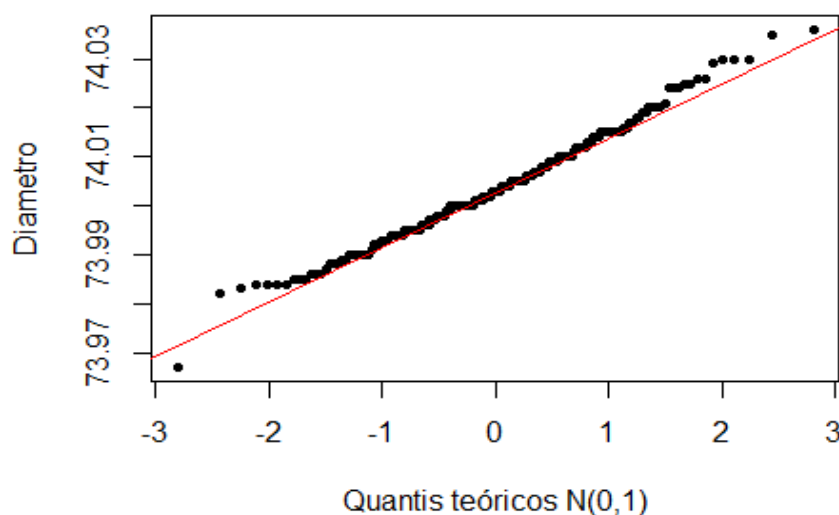



```
ggplot(Dados, aes (x = factor(trial), y=diameter, fill = trial))+
  geom_boxplot() + geom_boxplot(show.legend = T,
                                alpha = 0.5, outlier.color = "red")+
  theme_excel_new (base_size = 18)
```



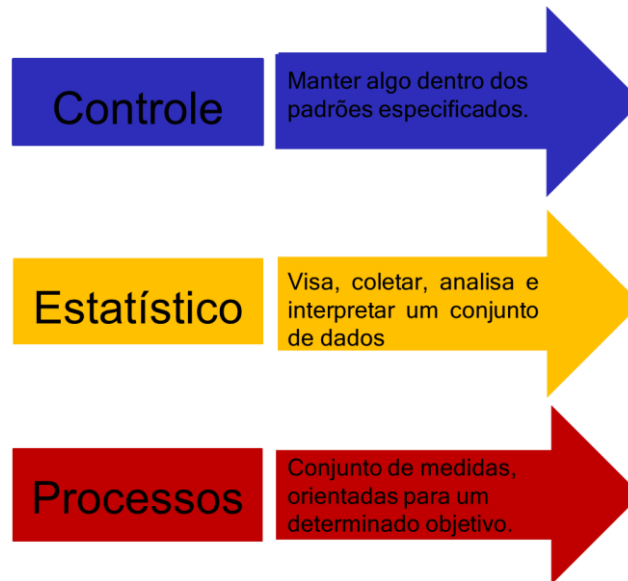
O gráfico quantil-quantil ou qq-plot, foi proposto por Wilk & Gnanadesikan, é usado na verificação de uma pressuposta de distribuição para um determinado conjunto de dados. O gráfico calcula valor teórico, desejado para cada ponto de dados baseando-se em uma distribuição definida. Os pontos, no gráfico se ajustarão formando aproximadamente uma linha, caso os dados seguirem a distribuição definida. No R, use a função *qqnorm()*.

```
qqnorm(Dados$diameter, main = "",
        xlab = "Quantis teóricos N(0,1)", pch = 20,
        ylab = "Diametro")
qqline(Dados$diameter, col = "red")
```



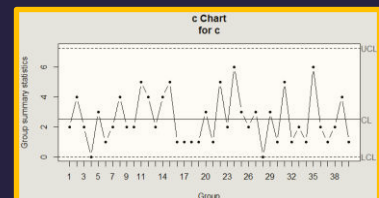
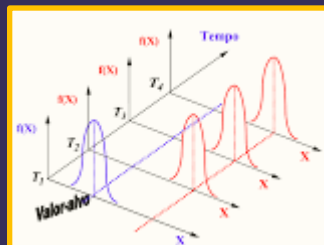
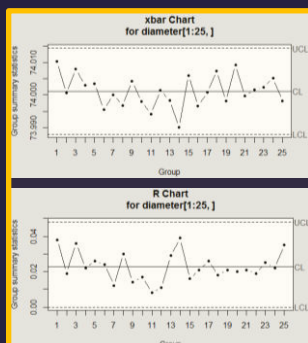
CONTROLE ESTATÍSTICO DE PROCESSOS (CEP)

O CEP preocupa-se em transformar os processos para que haja poucas variações, possibilitando melhorias na qualidade de produtos e serviços.



Cartas de controle

Entre as ferramentas de controle estatístico de qualidade, a carta de controle é preponderante, e busca avaliar se o processo mantém-se controlável e esperável, ou se existe exigências de ações para reparar o processo. Duas classes de cartas de controle são apresentadas, sendo para variáveis ou para atributos.



Cartas de controle para variáveis

As cartas de controle para variáveis, estão relacionadas com as características estimáveis em uma escala numérica, por exemplo, diâmetro, peso ou volume. O processo com característica variável, normalmente, é monitorado por duas cartas de controle, assim, a primeira carta monitora a centralidade e a segunda carta a dispersão. Aqui, serão mostrados a implementação das cartas \bar{X} e R que representa a média e amplitude, respectivamente.



Para formar as cartas, os parâmetros média e amplitude são utilizados, a média para centralidade (LM) e a amplitude para variabilidade. Além de determinar os limites superiores e inferiores (LSC e LIC). Os cálculos dos parâmetros são feitos através das equações abaixo.

LIMITES LSC E LIC PARA MÉDIA (\bar{X})

$$LSC_{\bar{X}} = \bar{X} + A_2 \cdot \bar{R}$$

$$LIC_{\bar{X}} = \bar{X} - A_2 \cdot \bar{R}$$

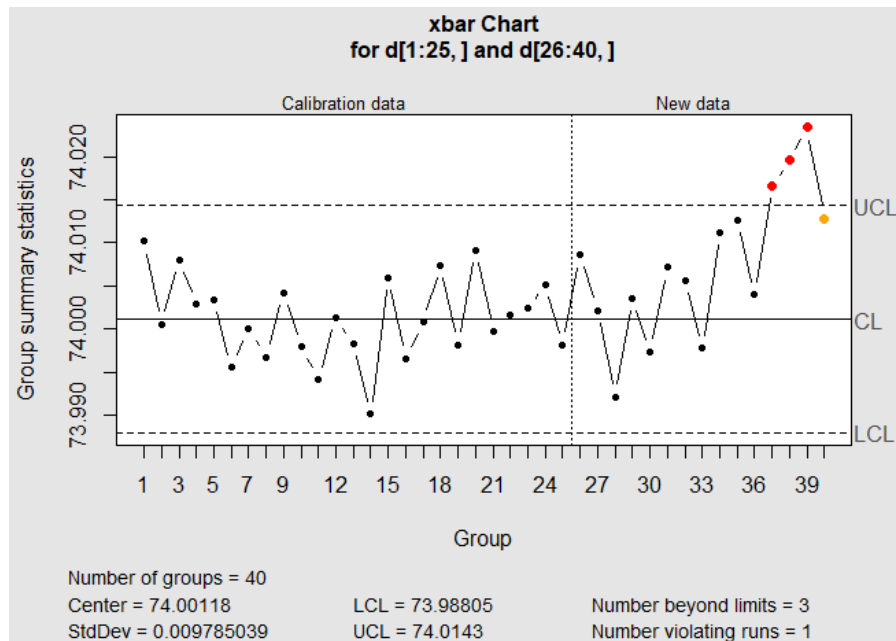
LIMITES LSC E LIC PARA AMPLITUDE R

$$LSC_R = D_4 \cdot \bar{R}$$

$$LIC_R = D_3 \cdot \bar{R}$$

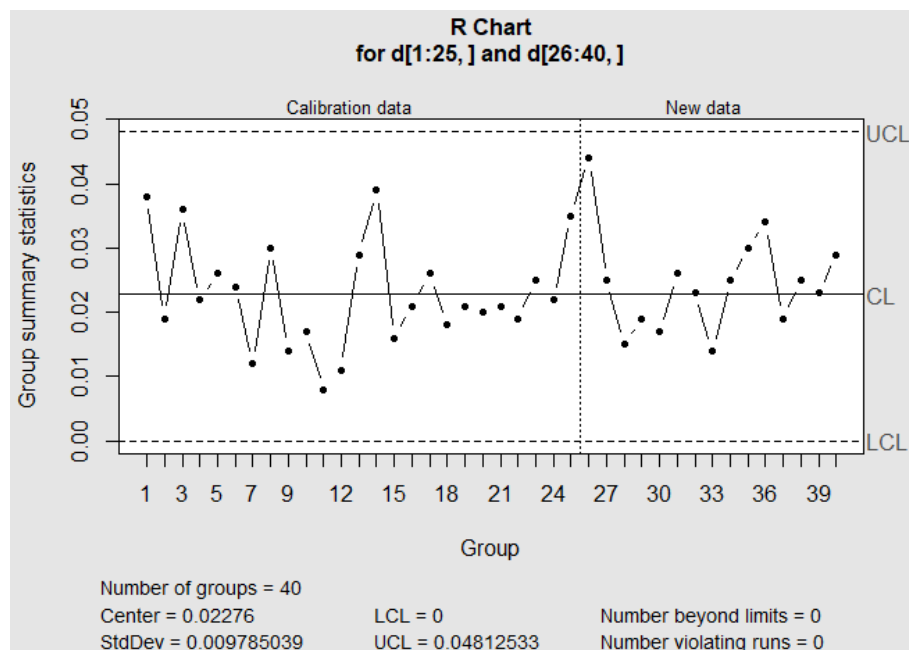
No R, a função `qcc()` gera o gráfico de controle Xbar para o conjunto de dados

```
grafico_Xbar= qcc(d[1:25,]), type = "xbar", newdata = d[26:40,])
```



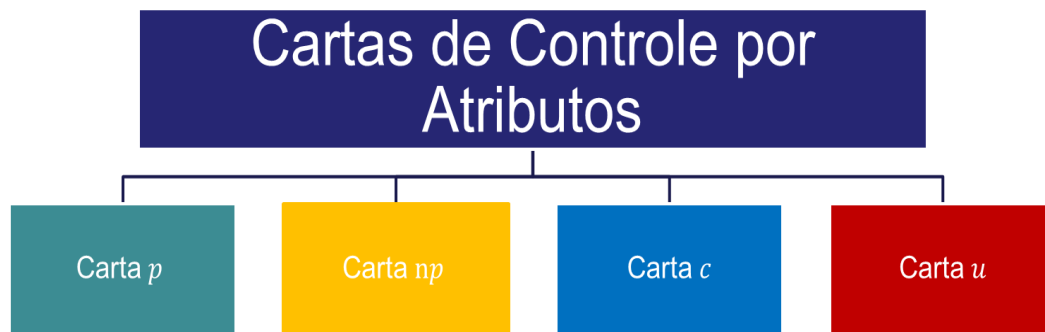
Alterando o argumento type da função qcc(), obten-se o gráfico para amplitude R.

```
grafico_Xbar= qcc(d[1:25,]), type = "xbar", newdata = d[26:40,])
```



Cartas de controle para atributos

A carte de controle para atributo, busca examinar características ditas como defeituosas e não defeituosas, conformes e não conformes etc. E são adequados considerando que há uma taxa de defeitos alta o suficiente para apresentar-se no gráfico de controle mediante uma amostra de comprimento coerente.



Para aplicação em R, neste mini curso, é usado a carta de controle para atributo p.

Tendo em conta a número de defeitos defeitos por subgrupos, a carta c é aplicada, em ocasião que todos os subgrupos tiverem o mesmo tamanho, em outras palavras, tiverem o mesmo número de itens, mediante as seguintes situações: Os defeitos se distribuem por um fluxo contínuo de certo produto, em que é possível estipular o número médio de defeitos; ou quando, encontra-se no grupo de amostras defeitos de origem e tipos distintos. Os cálculos dos parâmetros são feitos através das equações abaixo.

LIMITES LSC E LIC PARA carta c

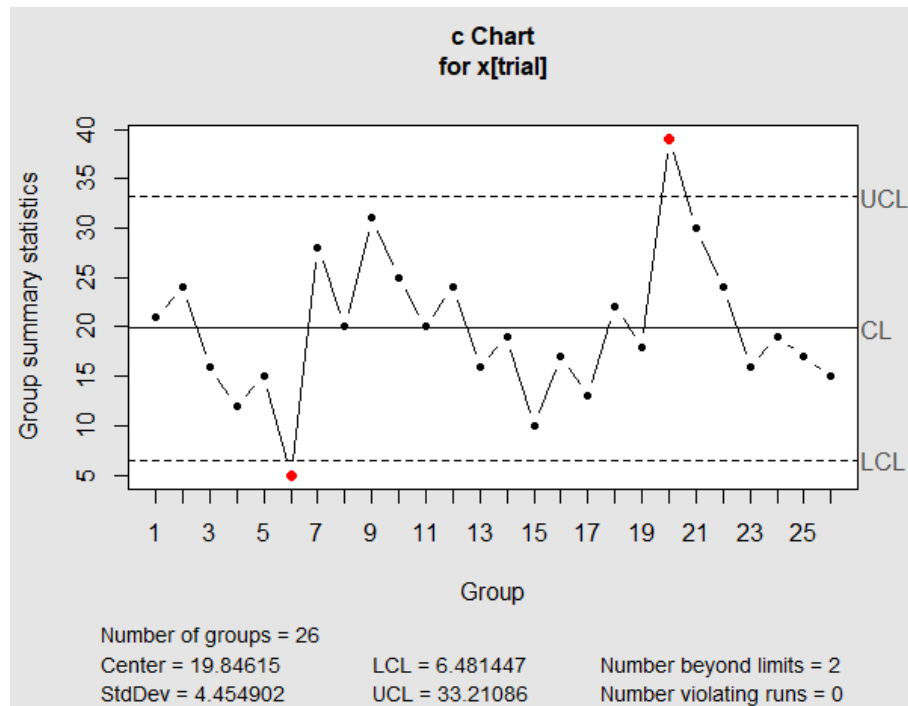
$$LSC = \bar{c} + 3\sqrt{\bar{c}}$$

$$LC = \bar{c}$$

$$LIC = \bar{c} - 3\sqrt{\bar{c}}$$

No R, a função `qcc()` gera o gráfico de controle apara atributo para o conjunto de dados.

```
grafico_Xbar= qcc(x[trial], sizes=size[trial], type = "c")
```



Capabilidade do processo

Análise de capacidade de processos visa, avaliar se um processo está estatisticamente controlado e de acordo com as especificações de engenharia. A análise de capacidade, é relevante para pressagiar até qual momento o processo se conservará dentro das tolerâncias, além de contribuir na triagem ou alteração de um processo, caracterizar condições de desempenhos de um determinado equipamento, elaborar o encadeamento de processo de produção caso exista efeitos interativos relativo as tolerâncias, e diminuir a variação presente em um processo de fabricação.

Considerando que as causas especiais de variação estão controladas, ou seja, o processo está sobre controle, é possível inteirar-se dos limites inerentes do processo.

Portanto, quando as medições resultarem em valores dentro dos limites especificados de acordo com o projeto, este processo, será dito capaz. Em casos opostos ao exposto, o processo será não capaz, nestas ocorrências existem indicativos estatístico da produção de produtos defeituosos.

Índice de capacidade (C_p)

O índice C_p , atenta-se a centralidade do processo, porém não indica necessariamente que as condições são atendidas, devido sua inclinação à dispersão. C_p pode ser definido como:

$$C_p = \frac{LSE - LIE}{6\sigma}$$

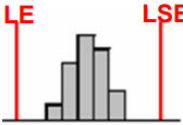
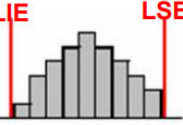
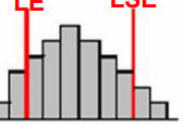
Onde:

LSE = Limite Superior de Especificação

LIE = Limite Inferior de Especificação

σ = desvio-padrão da amostra do processo

Na figura abaixo, é possível visualizar uma classificação de processos, em relação ao índice Cp.

CLASSIFICAÇÃO DO PROCESSO	COMPARAÇÃO DO HISTOGRAMA COM AS ESPECIFICAÇÕES	VALOR DE Cp
ADEQUADO CAPAZ		$C_p \geq 1,33$
ACEITÁVEL		$1 \leq C_p < 1,33$
INCAPAZ INADEQUADO		$C_p < 1$

Fonte: Adaptado de Werkema (1995, p.280)

Índice de desempenho do processo (Cpk)

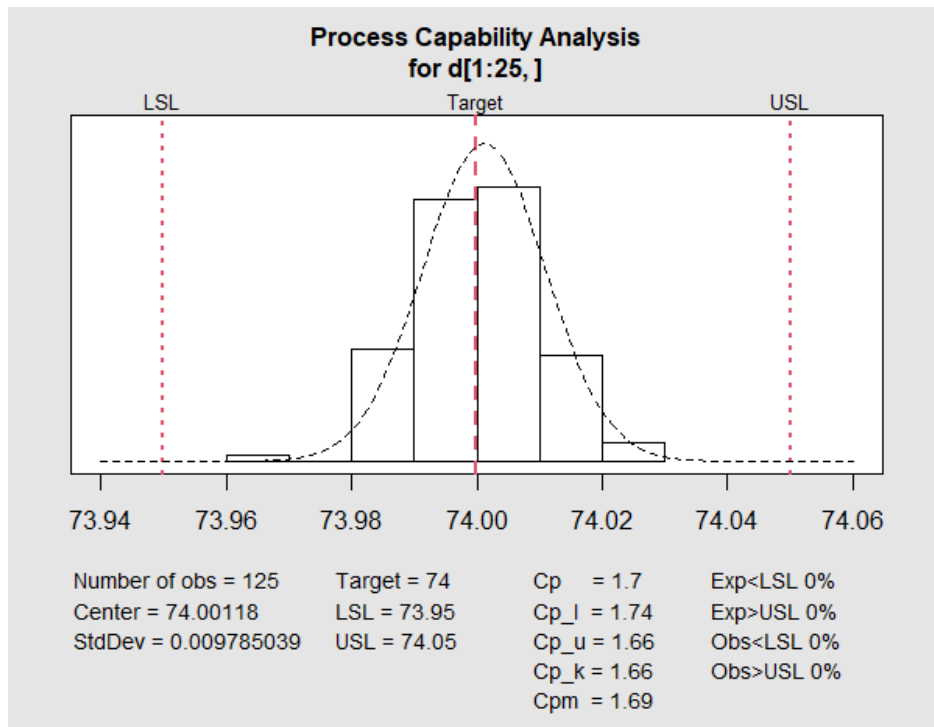
É o índice que expressa a capacidade real do processo, correspondendo a uma grandeza de posição e dispersão. As medidas de Cpk são estabelecidos pelas organizações, mas os valores de referências são usados normalmente conforme o quadro abaixo.

C_{pk}	INTERPRETAÇÃO DA CONFIABILIDADE	AÇÕES A PRATICAR	RELAÇÃO DO VALOR NOMINAL E À LINHA CENTRAL DO PROCESSO
$C_{pk} > 2,0$	ALTAMENTE CONFIÁVEL Processo Excelente	Operadores com perfeito controle do processo	Processo Centrado $C_p = C_{pk}$ Processo fora de alvo $C_{pk} \neq C_p$
$1,33 \leq C_{pk} \leq 2,0$	RELATIVAMENTE CONFIÁVEL Processo Capaz	Há necessidade dos operadores monitorarem para evitar deterioração.	Processo Centrado $C_p = C_{pk}$ Processo fora de alvo $C_{pk} \neq C_p$
$1,00 \leq C_{pk} \leq 1,3$	POUCO CONFIÁVEL Processo Relativamente Incapaz	Necessário controle contínuo pelos operadores	Processo fora do alvo, mas dentro dos limites de Especificação $C_{pk} < C_p$
$0 \leq C_{pk} < 1$	Processo Incapaz PODE-SE TER PRODUÇÃO DEFEITUOSA	Necessário controle de 100% da produção pelos operadores	Linha central do processo dentro ou coincidindo com um dos limites de Especificação (pode-se ter 50% de produção acima ou abaixo dos limites de Especificação) $C_{pk} < C_p$
$C_{pk} < 0$	NÃO TEM CONDIÇÕES DE MANTER AS ESPECIFICAÇÕES Processo Totalmente Incapaz	Necessário controle de 100% da produção pelos operadores	Linha central do processo fora dos limites de Especificação $C_{pk} < C_p$ Toda produção fora dos limites de especificação $C_{pk} < -1$

Fonte: Vieira (1999 apud Veit, 2003)

A função `process.capability()` descreve os índices de capacidade do processo.

```
capacidade = qcc(d[1:25,]), type = "xbar", nsigmas=3 plot= F)
process.capability (capacidade, spec.limits=c(173.95,74.05))
```



REFERÊNCIA

SAMOHYL, Robert Wayne, 1947- Controle estatístico da qualidade / Robert Wayne Samohyl. - Rio de Janeiro: Elsevier, 2009.

MORETTIN, Pedro Alberto, Estatística Básica/Pedro A. Morettin, Wilton O. Bussab. - 6. ed. - São Paulo : Saraiva, 2010.

MONTGOMERY, Douglas, C. Introduction to Statistical Quality Control - 6. ed. - Printed in the United States of America, 2009.

JUSTO, Luciana, Lopes. Implantação e Desenvolvimento do Controle Estatístico de Processos em uma Indústria Química: Uma Contribuição para o Desenvolvimento Local. 2009. 115 f. Dissertação (Mestrado em Gestão e Desenvolvimento Regional) - Universidade de Taubaté, Taubaté, 2009.

CORTIVO, Zaudir, Dal'. Aplicação do Controle Estatístico de Processo em Sequências Curtas de Produção e Análise Estatística de processo através do planejamento econômico. 2005. 151 f. Dissertação (Mestrado em Ciências) - Tecnologia da Universidade Federal do Paraná, Curitiba, 2005.