Lista de Exercícios Sobre R e Estatística Computacional

1 de Novembro de 2007

1 Programa

1. Análise de Dados: Cálculo de Momentos e Quantis, Estimação de Modelos de Regressão, Confecção de Histogramas e Gráficos.

2. Visão Geral sobre Diferentes Pacotes Estatísticos.

3. Introdução a Linguagens de Programação Interpretadas e Compiladas.

4. Simulação de Monte Carlo.

5. Aplicações de Estudo de Simulação de Monte Carlo à Estatística.

2 Introdução

O sistema R corresponde a um dialeto da liguagem S, que foi desenvolvida pela AT & T Bell Laboratories por Rick Becker. O R pode ser é obtido no site http://cran.r-project.org/.

Além de introduzir o R, estas notas irão revisar alguns algoritmos básicos da disciplina de programação 1 e introduzir alguns algoritmos de simulação e de modelagem estatística.

R é uma linguagem funcional que usa os mesmos símbolos do C, C++ e JAVA. As entidades em R, inclusive funções e estruturas de dados, podem ser operadas como dados.

2.1 Alguns Comandos básicos

A maior parte dos problemas propostos podem ser solucionados com a leitura do manual "An Introduction to R”, que esta disponível no próprio programa.

Por exemplo a leitura de um vetor direto do teclado pode ser feita por
x <- c(5, 6, 3.1, 6, 7).

Problema 1. *Como seria possível ler a matriz identidade de dimensão 2?*
Sugestão: Use o comando array

Matrizes de dados, que são o início das análises estatísticas, pode ser lidas diretamente do teclado. Para isto o comando data.frame( ) pode ser usado.

Problema 2. *Usando o comando data.frame(), entre com a matriz de dados peso=(60,70,80) e altura=(160,150,170).*

O R apresenta para vetores uma aritmética semelhante aquela utilizada para os escalares. Dessa forma, o comando

$$v < -2 * x + y + 1$$

esta correto, onde * significa o produto de um vetor por um escalar.

É muito fácil gerar sequências em R. O comando

$$1 : 30$$

produz uma sequência com 30 valores.

Problema 3. *Como gerar um vetor que tem 10 valores, com os valores em sequência de 1 a 10.*

Vetores lógicos estão disponíveis no R e assumem os valores TRUE, FALSE, and NA.

Os operadores lógicos são <, <=, >, >=, ==.

Estes operadores serão estudados com mais detalhes quando as estruturas de decisão forem estudadas.

Em alguns conjuntos de dados o valor de uma variável pode ser omisso. Neste caso, o R usar o símbolo NA. Por exemplo, o comando z <- c(1:5,NA) indica que a última observação esta omissa.

Uma outra possibilidade acontece quando um cálculo numérico não é possível. Neste caso, o R usa NaN para estes casos.

3 Estrutura de Decisão

Neste tópico serão desenvolvidos alguns pequenos programas em R. Estes programas representarão uma revisão da disciplina sobre programação e servirão de base para os programas de simulação e de modelagem estatística que serão desenvolvidos.

O principal comando desta seção é

$$if(cond)expr,$$

cond representa uma condição lógica e expr representa um ou mais comandos que serão executados.

O simples exemplo

```
x<-10
if(x<11) print("sai")
```

ilusta o uso do comando if.

Um exemplo do comando if, else corresponde a

```
x<-10
if(x<9) print("sai") else print("sai2")
```

Problema 4. *Fazer uma algoritmo em R que lê dois valores e imprime o maior.*

Problema 5. *Fazer um algoritmo que lê três valores e imprime o valor do maior.*

4 Matrizes, Vetores e Laços

Em R, as definições mais usadas de matrizes e vetores são array() e matrix().

Por exemplo,

$$dim(z) < -c(4, 5, 10)$$

é tambem uma forma de definir um vetor de dimensões 3, 5 e 10.

Outro exemplo o comando

$$vetor < -array(1 : 25, dim = c(5, 5))$$

define um matriz quadrada de dimensão 5, onde os elementos estão em sequência.

O comando

$$dados < -array(c(1, 2, 3, 4), dim = c(4, 1))$$

permite a leitura do vetor de dados (1,2,3,4).

Uma matriz quadrada, com todos elementos iguais a 1, de dimensão 10 é obtida pelo comando

$$A < -matrix(1, 10, 10)$$

Problema 6. *Dada uma lista com a nota de cinco alunos, por exemplo, {3,7,9,6,7} encontre a média, o desvio padrão e ordene as notas. Sugestão: usar as funções mean, sd, sort no vetor definido.*

É possível localizar uma linha ou uma coluna de uma matriz ou vetor de dimensão superior por usar os índices de forma apropriada. Por exemplo o elemento A[j,] corresponde a j-ésima linha da matriz A.

Se as matrizes A, B e C são compatíveis, então
D <- 2*A*B + C + 1
é adequada a sintaxe do R.

O produto de 2 matrizes A e B é definido por *A%*%B*. Um vetor a deve ser definido como uma matriz de uma coluna para posteriormente ser multiplicado pela matriz de interesse, por exemplo, *a%*%B*.

O transposto de uma matriz ou vetor pode ser obtido com a simples função *t()*. Por exemplo, B <- t(A).

A inversa de uma matriz A é dada pelo comando

$$solve(A)$$

A função eigen(A) calcula os autovalores e autovetores da matriz A. Com o comando

$$ev < -eigen(A)$$

os autovalores e autovetores da matriz A são atribuíduos a variável ev, que passa a conter todas estes resultado. Para obter os autovalores basta colocar ev$valeev$vec para obter os autovetores.

Similarmente ao mean, o R dispõe de comandos para calcula as somas, máximos e mínimos. Execute o seguinte programa:
A<-matrix(1:100,10,10)
sum(A[,2])
min(A[,1])
max(A[,1]).

Problema 7. *Dada uma lista com a nota de 10 alunos de duas turma, por exemplo, {3,7,9,6,7,6,8,9,4,9} e {3,6,9,6,7,7,8,8,4,8}. Encontrar um algoritmo que calcule quantos alunos possuem a mesma nota.*

A última questão pode precisar de uma estrutura de laço. o R dispõe de dois comandos

O comando *while* é um laço condicional, isto é, um comando é repetido enquanto um certa condição é satisfeita. Por exemplo, considere o algoritmo:

```
A<-matrix(1:100,10,10)
i<-1 j<-1
while(A[i,j]<50)
{ print(A[i,j])
i<-i+1 j<-j+1 }
```

O comando *for* é um laço cujo o número de repetições é fixado, o que denominado de laço contado. Isto contrasta com o while porque naquele caso o número de repetições não é fixado.

O exemplo abaixo ilustra o uso do *for* para a calcular potências de 2.

```
c<-2
for (i in 1: 3)
c<-c*2
```

Problema 8. *Fazer um algoritmo para calcular e imprimir as 10 primeiras potências de 3*

Problema 9. *Usar o comando for para calcular a soma de 10 termos de*

$$\exp^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \ldots$$

Problema 10. *Escreva um laço que calcula o fatorial de um inteiro n.*

Sugestão: usar o while.

Problema 11. *Um número é, por definição, primo se ele não tem divisores, exceto 1 e ele próprio. Prepara um algoritmo para ler um número e verificar se este é primo ou não.*

5 Funções Escritas pelo Usuário

O R permite que o usuário escreva funções de seu interesse. A definição geral de uma função é dada por

$nome < -$função$(arg_1, arg_2, ...)$expressão

Por exemplo, a função abaixo calcula o fatorial do número n.

```
fat<-function(n){
fat<-1
while(n!=1){
fat<-fat*n
n<-n-1 }
fat }
```

Exemplo: Escreva uma função para calcular $f(x) = x^2 - 3x + 2$. Atribua uma sequencia de 20 valores e faça o gráfico desta função.

```
f<-function(x) { x*x-3x+2
}
dados<-matrix(0,20,2)
dados[,1]<- -10:9
for(i in 1:20)
{
dados[i,2]<-f(dados[i,1])
}
plot(dados)
```

Problema 12. *Faça o gráfico do exemplo anterior com a função* $f(x) = x^3 + x^2 + x + 3$.

Problema 13. *Escreva funções para calcular a média e o desvio padrão de um vetor de valores. Compare os resultados de suas funções com aqueles resultados obtidos pelo R.*

Problema 14. *Escreva uma função que calcula o produtório.*

Problema 15. *Escreva uma função que calcula o valor de*

$$\pi = \sqrt{\sum_{i=1}^{\infty} \frac{6}{i^2}}$$

6 Distribuições de Probabilidades: Cálculos e Propriedades

Nesta seção iremos utilizar várias funções para gerar números aleatórios que estão implementadas no R. Porém, a compreensão destas funções será abordada nas próximas seções.

Por exemplo, a função *rnorm* é utilizada para gerar observações de uma variável com distribuição normal. Para ilustrar vários comandos, vamos verificar se o gerador de variáveis normais do R é adequado.

Os comandos abaixo são utilizados com k igual a 10, 100, 1000 e 10000.

```
x<-rnorm(k,0,1)
hist(x)
```

Problema 16. *Repita os histogramas supracitados com 10, 100, 1000 e 10000 observações de uma exponencial com média 1. Sugestão: use o comando rexp(n, rate = 1).*

Problema 17. *Com as 4 amostras da distribuição normal e as 4 amostras da distribuição exponencial, faça gráficos com o comando qqnorm, que fornece um gráfico do quantis da variável e os quantis da normal. Que conclusão você pode obter destes gráficos?*

Problema 18. *Coloque os títulos dos gráficos e os nomes das variáveis em português nos 8 gráficos que você fez nas duas questões anteriores.*

7 Geração de Números Aletórios

Antes dos computadores, números aleatórios eram gerados por retirar uma bola de uma urna, jogar um dado e outros métodos manuais.

É possível fazer um programa que gere números pseudo-aleatórios. Estes números não são considerados aleatórios porque são gerados por uma regra determinística. Porém, é possível verificar com testes estatísticos que estes números tem um comportamento aleatório.

Um dos métodos mais comuns de gerar números aletórios é iniciar com uma semente, por exemplo $x_0 = 2$, e sucessimente ir calculando a sequência de números aletórios com

$$x_n = ax_{n-1} \ \%\% \ m$$

onde a e m são constantes positivas e o símbolo $\%\%$ denota o resto da divisão inteira.

Problema 19. *Escreva um programa em R para implementar o gerador*

$$x_n = 3x_{n-1} \ mod \ 150.$$

Obtenham os 10 primeiros números desta sequência.

8 Método de Monte Carlo

Resumidamente, o método de Monte Carlo reproduz com a geração de números aleatórios a realização de um experimento. Este procedimento é repetido um número fixo de vezes para calcular a proporção de experimento bem sucedidos.

O método de Monte Carlo pode ser usado para calcular a integral

$$\theta = \int_0^1 g(x)dx.$$

O procedimento é simples. Gera-se k variáveis aleatórias uniformes (0,1) independentes $U_1, \ldots, U_k$. Calcula-se as variáveis aleatórias $g(U_1), \ldots, g(U_k)$. A média de $g(U_1), \ldots, g(U_k)$ converge para θ, isto é, para o valor da integral de interesse.

$$\sum_{i=1}^{k} \frac{g(U_i)}{k} \rightarrow E[(g(u)] = \theta \ \ quando \ \ k \rightarrow \infty$$

Por exemplo, o programa para calcular

$$\theta = \int_0^1 x^2 dx.$$

é dado por

```
U<-runif(1000)
GU<-U*U
theta<-mean(GU)
print(theta)
```

Problema 20. *Calcular, com o método de Monte Carlo, a integral*

$$\int_0^1 exp\{e^x\}dx.$$

Para calcular a integral

$$\theta = \int_a^b g(x)dx,$$

é necessário utilizar a transformação $y = (x - a)/(b - a)$, o que resulta em $dy = dx/(b - a)$ e

$$\theta = \int_0^1 g(a + [b - a]y)(b - a)dy.$$

Problema 21. *Calcule com o método de Monte Carlo a integral*

$$\int_{-2}^2 e^{x+x^2} adx.$$

Para calcular a integral imprópria

$$\theta = \int_o^\infty g(x)dx$$

usa-se a tranformação $y = 1/(x+1)$, o que implica em $dy = -dx/(x+1)^2 = -y^2dx$.

Problema 22. *Calcule com o método de Monte Carlo a integral*

$$\int_{-\infty}^\infty e^{-\frac{x^2}{2}} dx.$$

9 Cálculos de Probabilidade e Estatística com o Método de Monte Carlo

Vários problemas de probabilidade e estatística podem ser solucionados com simulação. Nesta seção serão apresentados alguns exemplos e problemas.

Suponha que um dado é lançado e deseja-se se estimar a probablidade do face observada ser 3. Apresentar um programa que calcule esta probabilidade

```
cont<-1 for(i in 1: 100)
face<-sample(6,1)
if (3==face) cont<-cont+1
print(cont/100)
```

Neste exemplo, o laço de Monte Carlo é utilizado para representar o lançamento de um dado. Assim, foram feitos 100 lançamentos devemos esperar o valor aproximado de 1/6.

O comando *sample* foi fundamental para o programa acima. Porém, é possível solucionar este problema de outra maneira.

```
cont<-1
for(i in 1: 100)
face<-round(6*runif(1))
if (3==face) cont<-cont+1
print(cont/100)
```

Problema 23. *Compare as duas soluções apresentadas nos exemplos anteriores. Qual é a mais apropriada? Explique porquê.*

Algoritmos de simulação podem ser usados para solucionar problemas de probabilidade.

Considere o problema: uma moeda honesta é jogada 3 vezes. Considere X o número de caras obtidas. Calcule a probabilidade de X igual a 2.

```
cont<-1
for(i in 1: 10000)
face1<-sample(c(0,1), 1, replace = TRUE)
face2<-sample(c(0,1), 1, replace = TRUE)
face3<-sample(c(0,1), 1, replace = TRUE)
face<-face1+face2+face3
if (3==face) cont<-cont+1
print(cont/10000)
```

A resposta deve ficar próxima de 1/8.

Problema 24. *Escreva um programa que apresenta um código mais simples para o exemplo anterior.*

É possivel simular uma variável com uma distribuição arbitrária. Se a distribuição é dada por

```
x<-array(10,4,20,15,6,7,12,14),dim=c(8,1)) hist(x)
```

Basta gerar uma variável aleatória uniforme e de acordo com o valor desta variável associar a um valor discreto de x.

Problema 25. *Um lote é formado por 20 peças defeituosas e 80 não-defeituosas. Se duas peças são escolhidas ao acaso sem reposição, qual a probabilidade de que ambas as peças sejam defeituosas?*

Problema 26. *Sabe-se que uma certa moeda apresenta cara três vezes mais frequente que coroa. Essa moeda é jogada três vezes. Seja X o número de caras que aparece. Estabeleça a função de probabilidade e a função de distribuição de X. Faça o gráfico de ambas.*

Problema 27. *Suponha que 5% de todas as peças de uma linha de fabricação sejam defeituosas. Se 10 destas peças forem escolhidas e inspecionadas, qual será a probabilidade de que no máximo duas defetuosas sejam encontradas.*

10 Método da Inversa

É possível utilizar a função de distribuição de uma variável aleatória contínua para gerar valores desta variável aleatória. Basta utilizar o fato de que uma variável aleatória contínua X com distribuição F pode ser gerada a partir da fórmula

$$X = F^{-1}(U),$$

onde U é uma variável uniforme.

Problema 28. *Encontre a distribuição da exponencial com média igual a 1. Escreva um algoritmo de simulaçao para gerar valores desta variável.*

Problema 29. *Escreva um algoritmo de simulação para gerar valores da variável aleatória X que tem distribuição $F(x) = 1 - exp(-\alpha x^{\beta}),\ \ 0 < x < \infty$. A distribuição de X é denominada de Weibull.*

11 Revisão

Problema 30. *Escreva funções no R para calcular as seguintes séries:*

$$\exp^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \ldots$$

$$\sin(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} \ldots$$

Problema 31. *Dado dois vetores quaisquer, ordenar estes vetores e obter um novo vetor sempre com o componente maior destes dois vetores.*

Problema 32. *Faça um algoritmo para multiplicar matrizes que seja compatíveis.*

Problema 33. *Suponha que o campeonato brasileiro tenha 5 times. Faça um programa que ler o resultado de cada partida e atualiza a saída, que é dada pelas variáveis Time, Jogo, Vitorias, Empates, pontos.*

Problema 34. *Gere uma matriz com 4 colunas e 20 linhas. Suponha que a médias das colunas são 20,30,40 e 50, respectivamente. Faça diagramas de dispersão entre as variáveis? O que voce observa? Use os comandos library(MASS) e mvrnorm(use help para ver detalhes deste comando) para gerar esta mesma matriz. Admita que existe uma matriz de covariância em mvrnorm. Quais as diferenças?*

Problema 35. *Use y<-rnorm(100) para gerar uma amostra aleatória da normal de tamanho 100. Calcule a média e o desvio padrão de y. Faça um loop e calcule as estatísticas supracitadas de y 100 vezes. Armazene os resultados da média em um vetor av. Calcule o desvio padrão de av e faça seu histograma. Comente o resultado.*

Problema 36. *Transforme o algoritmo anterior em uma função.*

Problema 37.