

BIS0005 - Bases Computacionais da Ciência

Aula 04 - Bases de Dados

Saul Leite

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

Q2 2018

Dados e Informação

Dados e informações são tratados como sinônimos, mas do ponto de vista do processamento computacional possuem significados **diferentes**.

Dados e Informação

Dados e informações são tratados como sinônimos, mas do ponto de vista do processamento computacional possuem significados **diferentes**.

- **Dado** (ou dado bruto): qualquer elemento que possa ser processado por um computador.
 - Conjunto de números;
 - Cadeias de caracteres;
 - Imagens;
 - Código de barras;
 - Senhas criptografadas;
 - (dentre outros)...
- **Informação**: a interpretação dada a um conjunto de dados, tornando-os significativos para algum contexto do mundo real.

Dados e Informação

Exemplo:

Em muitas livrarias, o processo de cobrança dos caixas é automatizado, realizado por meio de códigos de barras. O código de barras é uma sequência de barras verticais, que podem ser lidas por um scanner e carregadas em um sistema computacional. Como os códigos de barras podem ser manipulados computacionalmente, podemos chamá-los de **DADOS**.



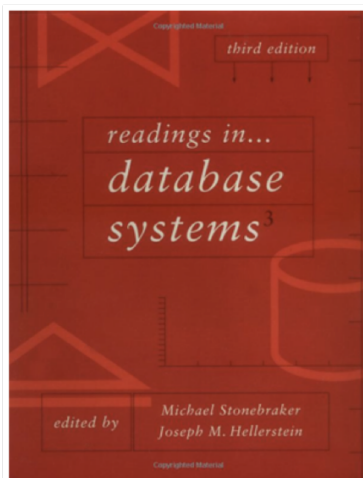
Dados e Informação

Apesar da forma válida de representação, um código de barras, **isoladamente**, tem pouco (ou nenhum) significado.

Dados e Informação

Apesar da forma válida de representação, um código de barras, **isoladamente**, tem pouco (ou nenhum) significado.

O código de barras pertence a um livro específico das prateleiras da livraria. Dessa forma, a **informação** representada pelo código é, nesse caso, o livro ao lado. Todas as vezes que o computador processar alguma operação relacionada ao código de barras, estará na realidade, processando a operação para o livro.



Bases de Dados

Bases de dados: termo utilizado para indicar um aglomerado, ou conjunto, de dados que deve ser manipulado pelo computador;

Pode ser definida como qualquer conjunto de dados para o qual se deseja

- **Armazenar**
- **Recuperar**
- **Visualizar**

conteúdo.

Bases de Dados

Exemplo (sem uso de computador): Lista de supermercado é uma base simples.

- **Armazenamento**
(persistência): permite uso futuro.
- **Recuperação**: acesso deve ser rápido e preciso.
- **Visualização**: forma de apresentação.



Bases de Dados

Exemplos de base de dados (sistemas computacionais): controle de clientes, correntistas, alunos, etc.

Características para grande volume de dados e grande quantidade de usuários:

- **Compartilhamento:** troca de informação entre usuários; acesso compartilhado aos dados. Segurança de infra-estrutura (garante funcionamento do hardware) e de acesso (controla permissões de usuários).
- **Regras de Consistência:** garantem coerência e qualidade dos dados; evitam valores inválidos.

Sistema Gerenciador de Bancos de Dados (SGBD)

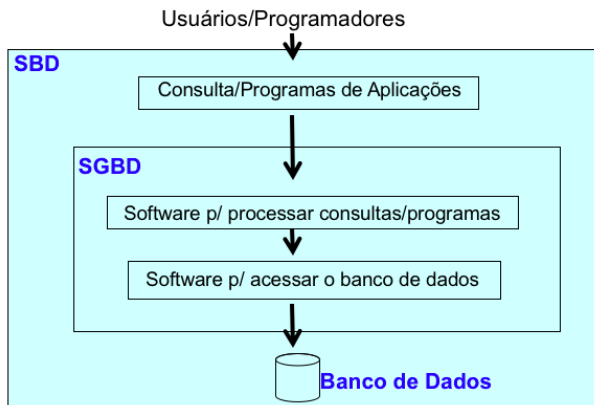
Sistema Gerenciador de Bancos de Dados (SGBD): São sistemas de propósito geral para a definição, criação e manipulação de Bancos de Dados.

- **Ex.:** Access, DB2, Oracle, SQLServer, BrOfficeBase, etc.

Sistema de Banco de Dados (SBD): “BD + SGBD + programas aplicativos”

- **Ex.:** sistemas de BD genômicos, sistemas bancários, telefônicos.

Sistema de Banco de Dados (SBD)



Objetivos: armazenar e recuperar informações com segurança, eficiência e flexibilidade

Tabelas

Existem vários modos de **organizar** uma base de dados: quando representada por uma planilha, a base de dados pode ser vista como uma matriz de dados:

- Facilita a visualização global dos dados.
- Identifica a posição de cada item de dado (linhas).
- Identifica os atributos dos dados (colunas).

É muito comum chamar a matriz de **tabela**.

Resultados de buscas e consultas em SGBD são apresentados em forma de **tabelas**.

Tabelas

Exemplo de uma **tabela**:

Produto	Fabricante	Qtd.	Medida	Preço	Supermercado
Suco	ValeSuco	1	l	3.00	Arpoador
Suco	Flash	1	l	4.50	Arpoador
Tomate	-	1	kg	3.50	Noite
Arroz	Tio José	5	kg	8.64	Noite
Arroz	Sem Broto	5	kg	9.99	Arpoador
Arroz	Da TV	1	kg	1.99	Noite
Feijão	Sem Broto	1	kg	4.00	Arpoador
Tomate	-	1	kg	2.99	Noite
Ovo	A Granja	12	u	3.19	Arpoador
Ovo	Caseiro	6	u	1.45	Noite
Suco	Flash	1	l	3.99	Noite

Organização de Dados

Linhas da tabela: armazenam **indivíduos** ou **itens** de dados.

Colunas da tabela: armazenam **atributos** (propriedades ou características) vinculados a cada item de dado armazenado nas linhas.

Produto	Fabricante	Qtd.	Medida	Preço	Supermercado
Suco	ValeSuco	1	l	3.00	Arpoador
Suco	Flash	1	l	4.50	Arpoador
Tomate	-	1	kg	3.50	Noite
Arroz	Tio José	5	kg	8.64	Noite

Tabela

Emp			
CódigoEmp	Nome	CodigoDepto	CategFuncional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	—

coluna (atributo)

nome do campo
(nome do atributo)

linha (tupla)

valor de campo
(valor de atributo)

- **Tabela:** conjunto de tuplas
- **Colunas:** atributos
- **Linhas:** tuplas, itens de dados ou instâncias
- **Campos:** valores dos atributos

Tipos de Dados

Toda a base de dados deve reservar uma especial atenção para a escolha dos **tipos de valores** armazenados (**tipos de dados**);

A escolha correta dos tipos de dados permite elevar o nível de qualidade da base:

- Controlar possíveis erros de cadastramento de valores;
- Previne inconsistências. Por exemplo:
 - evita o armazenamento de letras em locais destinados a valores numéricos;
 - evita datas escritas de forma incorreta.

Tipos de Dados

Dentre os tipos de dados destaca-se:

- **Número:** inteiro (1, 2, 3 ...), fracionário (1.23), etc.
- **Texto:** (cadeias de caracteres): “José Silva”.
- **Data:** 13/06/2011 (DD/MM/AAAA).
- **Hora:** 12:45:00 (hh:mm:ss).
- **Lógico:** VERDADEIRO, FALSO.

Além disso podemos ter **Domínios** para os atributos. Isso evita que valores inválidos sejam entrados. Exemplos:

- $\text{Dom}(\text{Med}) = \{\text{"kg"}, \text{"l"}, \text{"g"}, \text{"u"}, \text{"p"}\}$
- $\text{Dom}(\text{Qtd.}) = \{x \in \mathbb{Z} : x > 0\}$
- $\text{Dom}(\text{Preço}) = \{x \in \mathbb{R} : x \geq 0\}$

Arquivos Estruturados

Arquivos estruturados (ou flat files) são arquivos de dados organizados sob uma estrutura rígida e pré-definida:

- Os arquivos estruturados são muito usados para a troca de dados entre sistemas computacionais diferentes que não mantêm, entre si, um canal direto de comunicação;
- Definem-se caracteres (ou sequências de caracteres) que exercerão o papel de delimitadores de linha e de coluna.

Exemplo: Arquivo CSV (“Comma Separated Values”)

```
Produto;Fabricante;Qtd.;Medida;Preço;Supermercado;Data  
Suco;ValeSuco;1;1;3,00;Arpoador;12/05/2011  
Suco;Flash;1;1;4,50;Arpoador;12/05/2011  
Tomate; - ;1;kg;3,50;Noite;14/05/2011
```

Manipulação de Dados

Duas operações importantes para de recuperação (**Filtros**) e visualização (**Ordenação**) dos dados.

Filtros: é uma forma de recuperação de dados que consiste em realizar consultas sobre a base de dados, verificando valores e propriedades dos dados armazenados.

Ordenação: é uma operação comum quando se trabalha com dados. Dados ordenados facilitam a localização de informação e identificação de padrões. Além disso, a ordenação de dados pode ser considerada uma operação básica e presente em praticamente todos os sistemas que trabalham com visualização de dados.

Manipulação de Dados: Filtros

Filtros: consistem em realizar consultas sobre a base de dados, verificando valores e propriedades dos dados armazenados. Filtros são definidos através de um predicado lógico.

- Um predicado lógico é uma expressão que devolve um valor VERDADEIRO ou FALSO.
- O formato de um predicado é normalmente dado por:

<atributo> <operador> <valor>

em que:

- <atributo> é a característica usada como referência na consulta;
- <operador> é um **operador relacional** que devolve um resultado VERDADEIRO ou FALSO.

Manipulação de Dados: Filtros

operadores relacionais:

Operador	Descrição	Em R
<	menor	<
≤	menor ou igual	<=
>	maior	>
≥	maior ou igual	>=
=	igual	==
≠	diferente	!=

Ex.: Suponha que desejamos acessar somente os dados do produto arroz. Teríamos uma expressão como a seguinte (em R):

```
Produto == "Arroz"
```

Manipulação de Dados: Filtros

Operadores lógicos:

E: operador binário que constrói uma expressão com valor lógico (VERDADEIRO ou FALSO), combinando conjuntivamente dois predicados. (em R &)

OU: operador binário que constrói uma expressão com valor lógico, combinando dois outros predicados disjuntivamente. (em R |)

Exemplos:

```
Produto == "Arroz" & Preço < 2
```

```
Produto == "Arroz" | Supermercado == "Noite"
```

PARTE PRÁTICA

Tabelas no R

Tabelas na linguagem R são armazenadas em estruturas chamadas **Data Frames**.

Nós vamos usar a estrutura de **Data Frame** do pacote *tidyverse*.

- *tidyverse* é uma biblioteca do R que contém várias ferramentas para análise de dados (veja <https://www.tidyverse.org/>).

Para isso, vamos primeiro instalar essa biblioteca:

```
#Instala a biblioteca  
install.packages("tidyverse")
```

Você precisa chamar esse comando somente uma vez no seu computador.

Tabelas no R

Sempre que for usar a biblioteca *tidyverse* em uma nova sessão do R, você precisa carregar a biblioteca chamando:

```
#Carrega a biblioteca  
library(tidyverse)
```

Com a biblioteca carregada, podemos criar a nossa primeira *data frame* com o comando **tibble**:

```
#Cria uma data frame com colunas faltas e notas  
tibble( faltas = c(8,2,5,12,15,9,6),  
        notas = c(78,92,90,58,43,74,81))
```

```
## # A tibble: 7 x 2  
##   faltas notas  
##   <dbl> <dbl>  
## 1      8    78  
## 2      2    92  
## 3      5    90  
## 4     12    58  
## 5     15    43  
## 6      9    74  
## 7      6    81
```

Tabelas no R

Podemos atribuir as *data frames* em variáveis:

```
#Cria uma data frame com colunas faltas e notas  
# e atribui na variável alunos  
alunos <- tibble( faltas = c(8,2,5,12,15,9,6),  
                  notas = c(78,92,90,58,43,74,81))
```

Podemos visualizar o conteúdo da tabela contida no *data frame* usando o comando **View**, ou clicando no nome da variável *alunos* na aba superior direita chamada “environment” do RStudio.

```
#Visualiza a tabela  
View(alunos)
```

Tabelas no R

Podemos acessar as colunas da tabela usando o operador \$, da seguinte forma:

```
#Acessando a coluna de Faltas  
alunos$faltas
```

```
## [1] 8 2 5 12 15 9 6
```

```
#Acessando a coluna de Notas  
alunos$notas
```

```
## [1] 78 92 90 58 43 74 81
```

Note que as colunas são **vetores**.

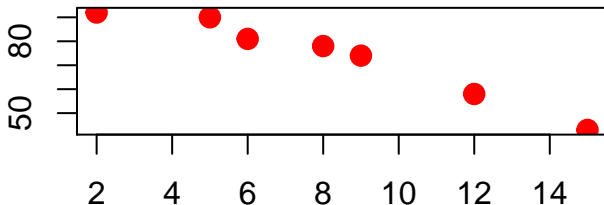
Tabelas no R

Ou seja, podemos usar essas colunas para fazer gráficos, calcular o coeficiente de correlação, calcular a média, mediana, variância e etc..

```
cor(alunos$faltas,alunos$notas)
```

```
## [1] -0.9747632
```

```
plot(alunos$faltas,alunos$notas,pch=19,col="red",  
      lwd=4, ylab = "", xlab= "")
```



Tabelas no R

Podemos também carregar tabelas escritas em formato CVS.

Para ilustrar esse processo, faça download do arquivo *compras.csv* que está no repositório do Tidial.

Podemos carregar essa tabela no R acessando, no canto superior direito, a aba **Environment** e clicando em **Import Dataset** → **From Text (readr)**.

Em seguida escolha o arquivo *compras.csv* e determine como delimitador (**delimiter**) o valor **semicolon** (ou seja, ponto e vírgula).

Após esse passo, acesse o menu **Locale** e altere o formato do ponto decimal (**decimal mark**) para vírgula ao invés de ponto.

Tabelas no R

Note que abaixo do nome de cada atributo, é informado o tipo de dado que compõe aquela coluna. Altere o tipo de dado da coluna **data** para o formato **Date** e escolha a formatação como sendo **%d/%m/%Y** (representando dia, mês, ano).

Import Text Data

File/Url:

Data Preview:

Produto (character)	Fabricante (character)	Qtd. (integer)	Medida (character)	Preço (double)	Supermercado (character)	Data (character)
Suco	ValeSuco	1	l	300	Arpoador	12/05/2011
Suco	Flash	1	l	450	Arpoador	12/05/2011
Tomate	-	1	kg	350	Nolte	14/05/2011
Arroz	Tio José	5	kg	864	Nolte	14/05/2011
Arroz	Sem Broto	5	kg	999	Arpoador	12/05/2011
Arroz	Da TV	1	kg	199	Nolte	14/05/2011
Feijão	Sem Broto	1	kg	400	Arpoador	12/05/2011
Tomate	-	1	kg	299	Nolte	14/05/2011
Ovo	A Granja	12	u	319	Arpoador	12/05/2011
Ovo	Caseiro	6	u	145	Nolte	14/05/2011
Suco	Flash	1	l	399	Nolte	12/05/2011

column 7: character

Previewing first 50 entries.

Import Options:

Name: ☒ First Row as Names

Skip: ☒ Trim Spaces

☒ Open Data Viewer

Delimiter:

Quotes:

Escape:

Comment:

NA:

Code Preview:

```
library(readr)
atividade1 <- read_delim("bases/atividade1.csv",
  ";", escape_double = FALSE, trim_ws = TRUE)
View(atividade1)
```

Reading rectangular data using readr


Tabelas no R

Note que os comandos usados para carregar a tabela estão no canto inferior direito, e são os seguintes para este arquivo:

```
library(readr)
compras <- read_delim("bases/compras.csv",
  ";", escape_double = FALSE,
  col_types=cols(Data = col_date(format = "%d/%m/%Y")),
  trim_ws=TRUE)
```


Tabelas no R

Durante a visualização da tabela podemos (1) **ordenar** as colunas para visualização e (2) fazer **filtros** e buscas.



The image shows a data table interface. At the top, there is a toolbar with icons for navigation and a 'Filter' button. To the right of the toolbar is a search bar with a magnifying glass icon. Below the toolbar, the table has columns: Produto, Fabricante, Qtd., Medida, Preço, Supermercado, and Data. Each of the first six columns has a dropdown menu with 'All' selected. The table contains 11 rows of data, numbered 1 to 11 in the first column. The data includes products like Arroz, Feijão, Ovo, Suco, and Tomate, along with their manufacturers, quantities, units, prices, and the supermarket they were purchased from.

	Produto	Fabricante	Qtd.	Medida	Preço	Supermercado	Data
1	Arroz	Tio José	5	kg	8.64	Noite	2011-05-14
2	Arroz	Sem Broto	5	kg	9.99	Arpoador	2011-05-12
3	Arroz	Da TV	1	kg	1.99	Noite	2011-05-14
4	Feijão	Sem Broto	1	kg	4.00	Arpoador	2011-05-12
5	Ovo	A Granja	12	u	3.19	Arpoador	2011-05-12
6	Ovo	Caseiro	6	u	1.45	Noite	2011-05-14
7	Suco	ValeSuco	1	l	3.00	Arpoador	2011-05-12
8	Suco	Flash	1	l	4.50	Arpoador	2011-05-12
9	Suco	Flash	1	l	3.99	Noite	2011-05-12
10	Tomate	-	1	kg	3.50	Noite	2011-05-14
11	Tomate	-	1	kg	2.99	Noite	2011-05-14

Obs.: Essas alterações são *somente para visualização* e não alteram os valores da tabela.

Tabelas no R: Criando Filtros

Podemos fazer filtros na linguagem R com o comando **filter**, que recebe como argumento um *data frame* e uma expressão lógica.

Exemplo:

```
filter(compras, Produto == "Arroz")
```

```
## # A tibble: 3 x 7
##   Produto Fabricante  Qtd. Medida Preço Supermercado Data
##   <chr>    <chr>      <int> <chr>    <dbl> <chr>      <date>
## 1 Arroz   Tio José        5 kg     8.64 Noite    2011-05-14
## 2 Arroz   Sem Broto       5 kg     9.99 Arpoador 2011-05-12
## 3 Arroz   Da TV           1 kg     1.99 Noite    2011-05-14
```

Tabelas no R: Criando Filtros

Exemplo: Selecionando produto arroz com quantidade maior ou iguais a 5kgs.

```
filter(compras, Produto == "Arroz" & Qtd. >=5)
```

```
## # A tibble: 2 x 7
```

##	Produto	Fabricante	Qtd.	Medida	Preço	Supermercado	Data
##	<chr>	<chr>	<int>	<chr>	<dbl>	<chr>	<date>
## 1	Arroz	Tio José	5	kg	8.64	Noite	2011-05-14
## 2	Arroz	Sem Broto	5	kg	9.99	Arpoador	2011-05-12

Tabelas no R: Criando Filtros

O Resultado do filtro pode ser atribuído a uma variável. No exemplo abaixo, o resultado da busca é armazenado na variável *arroz*:

```
arroz <- filter(compras, Produto == "Arroz")
arroz
```

```
## # A tibble: 3 x 7
##   Produto Fabricante   Qtd. Medida Preço Supermercado Data
##   <chr>    <chr>      <int> <chr>   <dbl> <chr>      <date>
## 1 Arroz    Tio José         5 kg     8.64 Noite    2011-05-14
## 2 Arroz    Sem Broto        5 kg     9.99 Arpoador 2011-05-12
## 3 Arroz    Da TV            1 kg     1.99 Noite    2011-05-14
```

Tabelas no R: Fazendo Ordenações

Podemos ordenar os elementos de uma tabela com o comando **arrange**, que recebe como argumento uma *data frame* e o nome de uma coluna para ordenação

```
arrange(arroz, Preço)
```

```
## # A tibble: 3 x 7
##   Produto Fabricante   Qtd. Medida Preço Supermercado Data
##   <chr>    <chr>      <int> <chr>   <dbl> <chr>      <date>
## 1 Arroz    Da TV           1 kg     1.99 Noite    2011-05-14
## 2 Arroz    Tio José        5 kg     8.64 Noite    2011-05-14
## 3 Arroz    Sem Broto       5 kg     9.99 Arpoador 2011-05-12
```

Obs.: O resultado do comando `arrange` é uma *data frame* e pode também ser armazenado em uma variável.

Tabelas no R: Fazendo Ordenações

Para ordenações decrescentes, podemos usar a função **desc**, da seguinte forma:

```
arrange(arroz, desc(Preço))
```

```
## # A tibble: 3 x 7
##   Produto Fabricante Qtd. Medida Preço Supermercado Data
##   <chr>    <chr>      <int> <chr>    <dbl> <chr>      <date>
## 1 Arroz    Sem Broto        5 kg      9.99 Arpoador  2011-05-12
## 2 Arroz    Tio José         5 kg      8.64 Noite   2011-05-14
## 3 Arroz    Da TV            1 kg      1.99 Noite   2011-05-14
```

Tabelas no R: Fazendo Ordenações

É possível também fazer ordenação por mais do que um atributo. No exemplo abaixo ordenamos a lista por Produto e Preço:

```
arrange(compras, Produto, Preço)
```

```
## # A tibble: 11 x 7
```

##	Produto	Fabricante	Qtd.	Medida	Preço	Supermercado	Data
##	<chr>	<chr>	<int>	<chr>	<dbl>	<chr>	<date>
##	1 Arroz	Da TV	1	kg	1.99	Noite	2011-05-14
##	2 Arroz	Tio José	5	kg	8.64	Noite	2011-05-14
##	3 Arroz	Sem Broto	5	kg	9.99	Arpoador	2011-05-12
##	4 Feijão	Sem Broto	1	kg	4	Arpoador	2011-05-12
##	5 Ovo	Caseiro	6	u	1.45	Noite	2011-05-14
##	6 Ovo	A Granja	12	u	3.19	Arpoador	2011-05-12
##	7 Suco	ValeSuco	1	l	3	Arpoador	2011-05-12
##	8 Suco	Flash	1	l	3.99	Noite	2011-05-12
##	9 Suco	Flash	1	l	4.5	Arpoador	2011-05-12
##	10 Tomate	-	1	kg	2.99	Noite	2011-05-14
##	11 Tomate	-	1	kg	3.5	Noite	2011-05-14

ATIVIDADE EM SALA

Exercício 1

Baixar o arquivo *Dados/crime.csv* do Tidia. Este arquivo contém informação sobre a região metropolitana de cidades americanas com dados da população (em milhares), densidade da população (por milha quadrada) e índice de criminalidade (crimes por cada 100.000 habitantes) (dados de 1968).

- Carregue a tabela no R;
- Descubra qual é a cidade mais populosa da tabela (usando o comando `arrange`);
- Encontre a cidade com maior índice de criminalidade (usando o comando `arrange`).
- Para cidades com população abaixo de um milhão de habitantes, veja qual atributo está mais correlacionado linearmente com o índice de crime (População ou Densidade).

Exercício 2

Baixar o arquivo *Dados/iris.csv* do Tidia. Este arquivo contém medições de comprimento e largura de pétalas e sépalas de três espécies de iris: virginica, versicolor, e setosa.

- Carregue a tabela no R;
- Calcule a média, mediana e variância para o comprimento da pétala para a espécie setosa.
- Faça uma regressão linear entre o comprimento da sépala e da pétala de iris para as espécies versicolor e virginica (juntas). Faça o gráfico de dispersão e o gráfico da reta de regressão.

Exercício 3

Baixar o arquivo *Dados/fertilidade.csv* do Tidia. Este arquivo contém dados sobre Fertilidade (Crianças por Mulher) e Contraceptivos (Percentual de anticoncepcionais entre mulheres casadas em idade fértil) para vários países de regiões diferentes do mundo.

- Carregue a tabela no R;
- Agrupe os dados baseados em Região, e calcule a média e variância dos atributos Fertilidade e Contraceptivos.
- Usando somente os dados da América Latina, meça o coeficiente de correlação entre Fertilidade e Contraceptivos. Faça um gráfico com a dispersão dos pontos e a reta de regressão.

Atividades para casa

Atividades para fazer até a próxima aula:

- Fazer todos os exercícios dos Slides (se ainda não fez).
- Fazer lista de exercícios no Tidia.
- Estudar para prova!

Prova

- Trazer cartão com dimensões 76,2mm X 127,0 mm.
- Podem anotar a **mão** qualquer informação ou comando para usar durante a prova.
- Anotações devem ser manuscritas **por você**.

Atenção: Qualquer outra situação (ex.: xerox do cartão do colega; cartões preparados pelo colega; ou cartões de dimensões maiores) será considerado **cola**.

Computador estará sem internet, **não** poderão usar qualquer outro dispositivo eletrônico além do computador do laboratório.

MATERIAL EXTRA

Tabelas no R: Agrupando dados

É possível agrupar dados com atributos iguais usando o comando **group_by** para fazer alguma medida estatística. Pode exemplo, suponha que desejamos agrupar os produtos de mesmo nome e calcular a média e desvio padrão dos seus preços:

```
#Passo1: Agrupa a tabela compras por Produto e salva  
# o resultado em uma nova tabela chamada agrupado  
agrupado <- group_by(compras, Produto)  
agrupado
```

```
## # A tibble: 11 x 7  
## # Groups:   Produto [5]  
##   Produto Fabricante Qtd. Medida Preço Supermercado Data  
##   <chr>    <chr>      <int> <chr>    <dbl> <chr>      <date>  
## 1 Suco     ValeSuco        1 l        3   Arpoador  2011-05-12  
## 2 Suco     Flash          1 l       4.5   Arpoador  2011-05-12  
## 3 Tomate   -              1 kg       3.5   Noite     2011-05-14  
## 4 Arroz    Tio José       5 kg      8.64   Noite     2011-05-14  
## 5 Arroz    Sem Broto      5 kg     9.99   Arpoador  2011-05-12
```

Tabelas no R: Agrupando dados

Já que agrupamos os dados por produtos, vamos agora calcular a média e variância dos seus preços usando o comando **summarise**:

```
#Passo 2: Faz medidas estatísticas com os dados agrupados  
summarise(agrupado, "Média Preço" = mean(Preço),  
           "Desvio Padrão" = sd(Preço))
```

```
## # A tibble: 5 x 3  
##   Produto `Média Preço` `Desvio Padrão`  
##   <chr>      <dbl>         <dbl>  
## 1 Arroz      6.87           4.28  
## 2 Feijão     4              NA  
## 3 Ovo        2.32           1.23  
## 4 Suco        3.83           0.763  
## 5 Tomate     3.24           0.361
```


Referências

- Aulas dos Profs. David Correa Martins Jr, Wagner Tanaka Botelho, Jesús P. Mena-Chalco.
- Livro Bases Computacionais da Ciência.