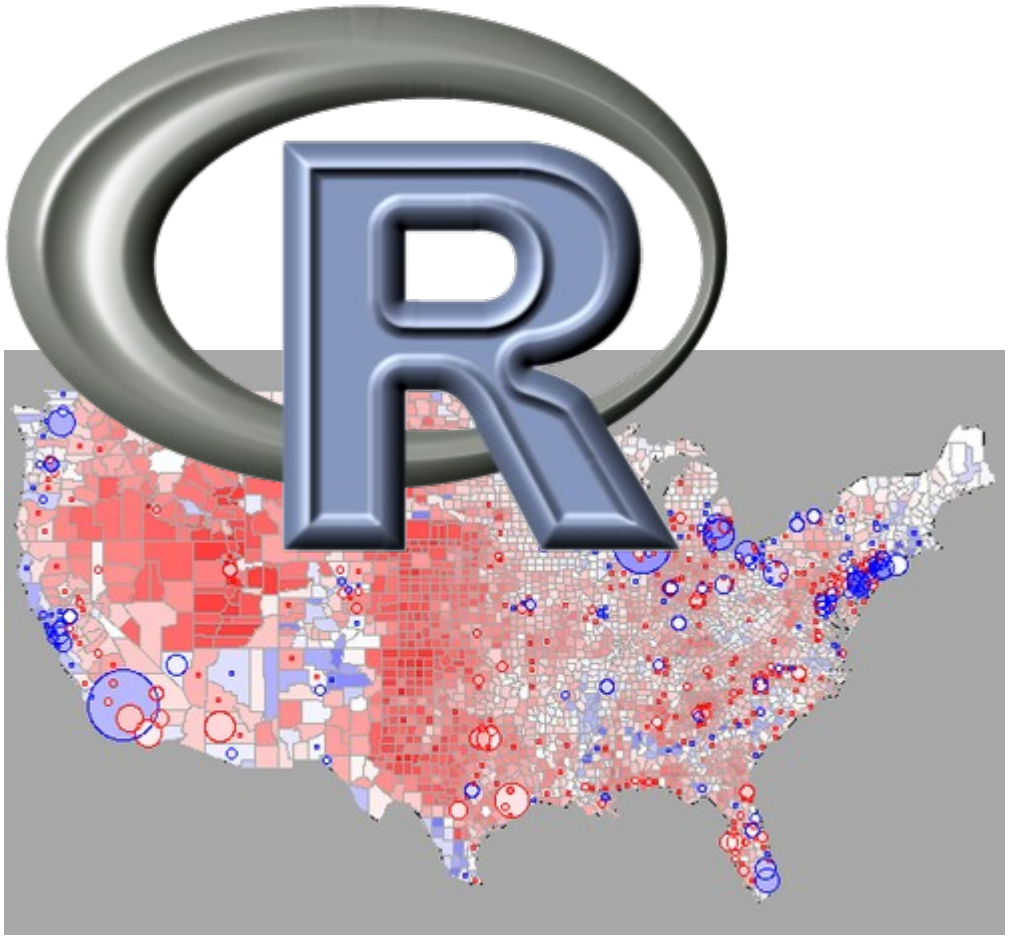
FIAP

PÓS-GRADUAÇÃO



MBA em Big Data

Introdução à Linguagem R

Encontro 2

Prof. Antonio Henrique Pinto Selvatici

antoniohps@gmail.com

Versão 1 – 10/2014

Professor

- **Antonio Henrique Pinto Selvatici**

- É engenheiro eletrônico formado pelo Instituto Tecnológico de Aeronáutica (ITA), com mestrado e doutorado pela Escola Politécnica (USP), e passagem pela Georgia Institute of Technology em Atlanta (EUA). Desde 2002, atua na indústria em projetos nas áreas de robótica, visão computacional e internet das coisas, aliando teoria e prática no desenvolvimento de soluções baseadas em Machine Learning, processamento paralelo e modelos probabilísticos, tendo desenvolvidos projetos para Avibrás, Rede Globo, IPT e Systax. Foi professor do curso de Ciência da Computação da Uninove de 2009 a 2013. Em 2012, tendo ajudado a fundar a Selsantech, participou do desenvolvimento do CatSearch, uma solução para Data Mining preparada para o paradigma MapReduce. É professor do MBA do curso de Big Data da FIAP e trabalha na reformulação do sistema de pagamento on-line eWally.

Resolução da lista 1

- Baixar o pacote de exercícios de:
  - https://dl.dropboxusercontent.com/u/22050262/Lista1-LinguagemR.zip
  - Arquivos extraídos para o diretório inicial do R
- Importar ResultadosRTLS.csv
  - `data <- read.csv('ResultadosRTLS.csv')`
- Plotar as colunas LM-FT,MCMC-LM-FT (gráfico de linhas)
  - `plot(data$LM.FT, type="l")`
  - `plot(data$MCMC.LM.FT, type="l")`
- Mostrar os índices em que a coluna LM é maior do que MCMC-LM
  - `which(data$LM > data$MCMC.LM)`
- Mostrar as médias das coluna LM e MCMC-LM
  - `mean(data$LM)`
  - `mean(data$MCMC.LM)`

Resolução da lista 1

- Importar COTAHIST.A1997 no formato FWF. Usar as informações do arquivo SeriesHistoricas_Layout.pdf para definir o formato de importação. Limite o número de entradas lidas para 10000.
  -
```
data <- read.fwf("COTAHIST.A1997", strip.white=T,
widths=c(2,8,2,12,3,12,10,3,4,13,13,13,13,13,13,13,5,1
8,18,13,1,8,7,13,12,3), header=FALSE, skip=1,
col.names=c("TIPREG", "DATA", "CODBDI", "CODNEG",
"TPMERC", "NOMRES", "ESPECI", "PRAZOT", "MODREF",
"PREABE", "PREMAX", "PREMIN", "PREMED", "PREULT",
"PREOFC", "PREOFV", "TOTNEG", "QUATOT", "VOLTOT",
"PREEXE", "INDOPC", "DATVEN", "FATCOT", "PTOEXE",
"CODISI", "DISMES"), n=20000)
```

Resolução da lista 1

- Fazer o gráfico “Scatter Plot” entre os valores de fechamento de duas ações à sua escolha.
  - `acao1 <- "ACE 4"`
  - `acao2 <- "ARN 4"`
  - `indices1 = data$CODNEG==acao1 & data$CODBDI=="2";`
  - `indices2 = data$CODNEG==acao2 & data$CODBDI=="2";`
  - `datas1 <- data$DATA[indices1]`
  - `datas2 <- data$DATA[indices2]`
  - `datas <- intersect(datas1,datas2)`
  - `valores1<-data$PREULT[indices1 & data$DATA %in% datas]`
  - `valores2<-data$PREULT[indices1 & data$DATA %in% datas]`
  - `plot(valores1,valores2)`

Resumo

- Gráficos em R
  - Tipos básicos de gráficos
  - Embelezando: pacote `ggplot2`
  - Modificando os elementos do gráfico
- Autoria e relatórios em R
  - R Markdown
  - Pacotes de autoria
- Lista de exercícios

2. Construção de Gráficos em R

1 – Visualização de dados básica

- A linguagem R possui muitos recursos para visualização gráfica
- Além dos gráficos padrão da linguagem R, há pacotes específicos para a geração de gráficos
- Sendo uma linguagem voltada para estatística, além dos gráficos convencionas o R fornece um modo fácil de criar representações estatísticas
- Vamos fazer uma comparação entre a plotagem básica, plotagem com o pacote `ggplot2` (o mais popular)
- Há o pacote `ggvis`, o sucessor do ggplot2, mas ainda muito incipiente
- Como instalar um pacote:
  - Packages -> Install (digitar o nome do pacote)
  - `install.packages("<nome do pacote>")`
- Instalando `ggplot2` (já está instalado no RStudio)
  - `install.packages("ggplot2")`
- Habilitando um pacote:
  - Tickar no box ao lado do nome do pacote
  - `library("<nome>")`

2 – Gráfico de dispersão (scatter plot)

- Esse tipo de gráfico mostra como o valor de uma variável influencia em outra
- Vamos usar os dados do data frame `mtcars`
  - `data()` mostra os data sets disponíveis no R
- Usando o comando básico
  - `plot(mtcars$wt, mtcars$mpg)`
- Usando `ggplot2`
  - `qplot(mtcars$wt, mtcars$mpg)`
  - `qplot(wt, mpg, data=mtcars)` #os dois vetores estão no mesmo data frame
  - `ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()`

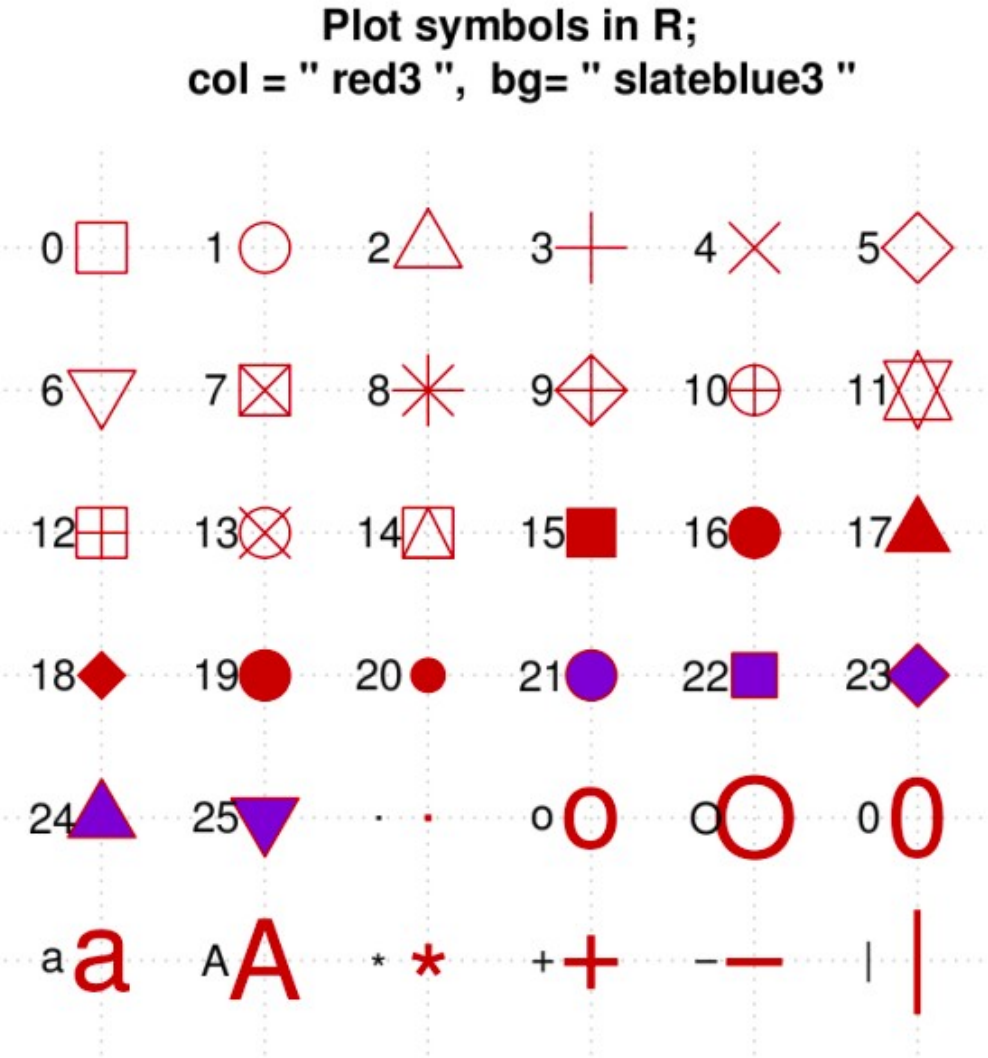
3 – Opções gráficas

- Tanto os comandos básicos de plotagem quanto os comandos de `ggplot2` permitem a inserção de opções de gráficas, como a nomeação dos eixos, título do gráfico, cores, etc.
- Título do gráfico
  - `plot(mtcars$wt, mtcars$mpg, main="Peso vs. Consumo")`
  - `qplot(wt, mpg, data=mtcars)+ggtitle("Peso vs. Consumo")`
- Rótulos dos eixos
  - `plot(mtcars$wt, mtcars$mpg, xlab="Peso", ylab="Consumo")`
  - `qplot(wt, mpg, data=mtcars, xlab="Peso", ylab="Consumo")`
- Limites dos eixos
  - `plot(mtcars$wt, mtcars$mpg, xlim=NULL, ylim=c(0,30))`
  - `qplot(wt, mpg, data=mtcars, xlim=c(0,NA), ylim=c(0,30))`
- Escala logaritmica
  - `plot(mtcars$wt, mtcars$mpg, log="y")`
  - `qplot(wt, mpg, data=mtcars, log="y")`

3 – Opções Gráficas >> Símbolos e Cores

- Cor do gráfico
  - `plot(mtcars$wt, mtcars$mpg, col="red")`
  - `qplot(wt, mpg, data=mtcars) + geom_point(color="blue")`
  - `qplot(wt, mpg, data=mtcars, colour=cyl) + scale_color_gradient(low = "blue", high="red")`
- Tipo e tamanho do símbolo
  - Alguns símbolos possuem cor de frente (`fg`) e cor de fundo (`bg`)
  - `with(mtcars,plot(wt,mpg,pch=22,cex=cyl/max(cyl),bg="yellow"))`
  - `qplot(wt, mpg, data=mtcars, size=cyl) + geom_point(shape='*')`
- Cor de fundo
  - `par(bg="green") #valor default: par(bg=NA)`
  - `plot(mtcars$wt, mtcars$mpg)`
  - `qplot(wt, mpg, data=mtcars, size=cyl) + theme(plot.background=element_rect(fill = "green"))`

3 – Opções Gráficas >> Símbolos e Cores



4 – Gráfico de linhas

- Esse tipo de gráfico mostra a evolução de uma variável com a outra (mais usada em funções deterministas)
- Vamos usar os dados do data frame `pressure`
- Usando o comando básico
  - `plot(pressure$temperature, pressure$pressure, type="l")`
- Usando `ggplot2` (`geom` é a opção de gráfico)
  - `qplot(pressure$temperature, pressure$pressure, geom="line")`
  - `qplot(temperature, pressure, data=pressure, geom="line")`
  - `ggplot(pressure, aes(x=temperature, y=pressure)) + geom_line()`

4 – Gráfico de linhas >> mais linhas e pontos

- Para acrescentar pontos ao gráfico construído com `plot`
  - `plot(pressure$temperature, pressure$pressure, type="l")`
  - `points(pressure$temperature, pressure$pressure)`
  - `lines(pressure$temperature, pressure$pressure/2, col="red")`
  - `points(pressure$temperature, pressure$pressure/2, col="red")`
- Usando `ggplot2`
  - `qplot(temperature, pressure, data=pressure, geom=c("line", "point"))`
  - `ggplot(pressure, aes(x=temperature, y=pressure)) + geom_line() + geom_point()`
- Mais de uma linha: devemos ter uma terceira variável dentro do data frame para agrupar os valores a serem plotados em linhas diferentes

4 – Gráfico de linhas >> mais linhas e pontos

- Acrescentando novos gráficos com `ggplot2` através de `melt`
  - `library("reshape2")`
  - `d1<-with(pressure, data.frame(temperature, pressure1=pressure, pressure2=pressure/2))`
  - `d <-melt(d1,id.vars = "temperature")`
  - `ggplot(d, aes(x=temperature, y=value, colour=variable, group=variable)) + geom_line() + geom_point() + scale_color_manual(values=c("blue","red"))`

- A função `melt` agrupa os valores das variáveis (colunas) de medidas em uma única coluna (`value`), parametrizada pelo valor de `id` (eixo x) fornecido correspondente e pelo nome da variável medida (coluna `variable`)
- A função `with` executa outras funções considerando criando um ambiente node as colunas de um data frame tornam-se vetores simples, disponíveis para serem usados

4 – Gráfico de linhas >> empilhando áreas

- Usar geom_area() e especificar as cores do preenchimento
  - `ggplot(d, aes(x=temperature, y=value, colour=variable, fill=variable, group=variable)) + geom_area() + scale_color_manual(values=c("blue","red")) + scale_fill_manual(values=c("blue","red"))`

- Especificando a ordem das linhas
  - `ggplot(d, aes(x=temperature, y=value, colour=variable, fill=variable, group=variable, order=rev(variable))) + geom_area() + scale_color_manual(values=c("blue","red")) + scale_fill_manual(values=c("blue","red"))`

5 – Gráfico de barras

- Um gráfico de barras mostra valores na forma de barras verticais
- Comando básico
  - `barplot(mtcars$mpg,names.arg=rownames(mtcars))`
- Com ggplot2
  - `qplot(rownames(mtcars), mtcars$mpg, geom="bar", stat="identity")`
  - Ordenando os rótulos de acordo com a ordem original
    - `carnames <- factor(rownames(mtcars), levels=rownames(mtcars))`
    - `qplot(carnames, mtcars$mpg, geom="bar", stat="identity")`
- Rotacionando os rótulos das barras:
  - `barplot(mtcars$mpg,names.arg=rownames(mtcars),las=2)`
  - `qplot(rownames(mtcars), mtcars$mpg, geom="bar", stat="identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1))`

5 – Gráfico de barras >> gráfico comparativo

- Plotando barras empilhadas
  - `barplot(rbind(mtcars$disp, mtcars$hp), names.arg=rownames(mtcars), las=2, legend=c("disp","hp"))`
  - `d1<- data.frame(car=factor(rownames(mtcars)), disp=mtcars$disp,hp=mtcars$hp)`
  - `d <- melt(d1,id.vars = "car")`
  - `qplot(car, value,data=d,colour=variable,fill=variable, group=variable, geom="bar", stat="identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1))`
- Plotando barras lado a lado
  - `barplot(rbind(mtcars$disp, mtcars$hp), names.arg=rownames(mtcars), las=2, legend=c("disp","hp"),beside=TRUE)`
  - `qplot(car, value,data=d,colour=variable,fill=variable, group=variable, geom="bar", position = "dodge", stat="identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1))`

6 – Gráfico de pontos (dot chart)

- Comando básico
  - `dotchart(mtcars$mpg,labels=row.names(mtcars))`
- ggplot2
  -
```
qplot(rownames(mtcars), mtcars$mpg, geom="point") +
coord_flip() + theme( #remove the vertical grid lines
      panel.grid.major.x = element_blank(),
#explicitly set the horizontal lines
panel.grid.major.y=element_line(linetype=3,color="darkgray"),
axis.text.y=element_text(size=rel(0.8)) )
```

7 – Gráfico de pizza

- Indica graficamente certa característica dentro de uma população
- Preparação dos dados
  - `d <- data.frame(HairEyeColor)`
  - ```
    fs <- data.frame(freq = with( d,
    c(sum(Freq[Hair=="Black"]), sum(Freq[Hair=="Brown"]),
    sum(Freq[Hair=="Red"]), sum(Freq[Hair=="Blond"]))))
    ```
  - `fs$cor=factor(c("Preto","Castanho","Ruivo","Loiro"))`
- Comando básico
  - `pie(fs$freq,labels=fs$cor,col=rainbow(4))`
- ggplot2
  - ```
    qplot(factor(""),freq,data=fs,fill=cor) +
    geom_bar(stat="identity") + coord_polar(theta = "y") +
    scale_x_discrete("") +
    scale_color_manual(values=rainbow(4)) +
    scale_y_continuous(breaks = cumsum(fs$freq)-fs$freq/2,
    labels = fs$freq)
    ```

8 – Combinando Múltiplos Gráficos

- Usar as funções `par()` ou `layout()`
- Com `par()`, a opção `mfrow=c(nrows, ncols)` cria uma matriz de `nrows` por `ncols` gráficos preenchda por linha. Com `mfcol=c(nrows, ncols)`, a matriz é preenchida por colunas.
  - `attach(mtcars)`
  - `par(mfrow=c(2,2))`
  - `plot(wt,mpg, main="Dispersão de peso vs. rendimento")`
  - `plot(wt,disp, main="Dispersão de peso vs cilindradas")`
  - `barplot(mpg,names.arg=rownames(mtcars), las=2, main="Comparativo de consumo")`
  - `barplot(disp,names.arg=rownames(mtcars), las=2, main="Comparativo de cilindradas")`
  - `detach(mtcars)`

8 – Combinando Múltiplos Gráficos >> layout()

- Com `layout()`, deve-se fornecer uma matriz de inteiros onde valores iguas são considerados como sendo a extensão do espaço do gráfico.
- Um gráfico na primeira linha e dois na segunda
  - `attach(mtcars)`
  - `layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))`
  - `barplot(wt,names.arg=rownames(mtcars),las=2,main="Comparativo de peso")`
  - `barplot(mpg,las=2,main="Comparativo de consumo")`
  - `barplot(disp,las=2,main="Comparativo de cilindradas")`
  - `detach(mtcars)`

8 – Combinando Múltiplos Gráficos >> layout()

- A primeira linha ocupa 2/5 da altura gráfico, deixando o resto para a segunda
- A primeira coluna ocupa 2/3 da largura do gráfico
  - `attach(mtcars)`
  - `layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE), widths=c(2,1), heights=c(2,3))`
  - `barplot(wt,names.arg=rownames(mtcars),las=2,main="Comparativo de peso")`
  - `barplot(mpg,las=2,main="Comparativo de rendimento")`
  - `plot(disp,mpg)`
  - `detach(mtcars)`

3. Criando documentos – Markdown do R

1 – O que é o Markdown

- O Markdown é um formato de edição para a criação de documentos, apresentações e relatórios a partir dos resultados do R.
- Possui uma sintaxe própria de formatação, e pode embarcar código R, mostrando seu resultado
- Criar um arquivo de markdown dentro do Rstudio é um bom começo de aprendizado
- Formatação básica:
  - Negrito: `**bold expression**`
  - Itálico: `*italic expression*`
  - Título principal: uma expressão com “=” na linha de baixo
    - `Meu título principal`
    - `==========`
  - Título secundário: uma expressão com hífens “-” na linha de baixo
    - `Meu título secundário`
    - `----------`
  - Código inline: `` `isso eh cohdigo do R` ``

2 – Blocos de texto do Markdown

- Metadata: informações sobre a autoria e o formato de saída do documento.

```
---
title: "Sample Document"
output:
    html_document:
        toc: true
        theme: united
---
```

- Listas:

```
- Primeiro item
    - Primeiro subitem
- Segundo item
- Terceiro item
```

2 – Blocos de texto do Markdown

- Bloco de código

```
```

Tudo o que estiver dentro dessas linhas será considerado código para fins de formatação

```
```

- Listas:

```
- Primeiro item
    - Primeiro subitem
- Segundo item
-Terceiro item
```

- Citação:

```
> Essa é uma citação por J. R. R. Rolemberg ...
```

- Equações no formato Latex
  - *$equation$* para equalções inline: `$\frac{d\exp(x)}{dx}=\exp(x)$`
  - *$$ equation $$* para equações destacadas: `$$\frac{d\exp(x)}{dx}=\exp(x)$$`
  - *$$* para equações MathML também funciona.

3 – Inserindo código R

- Código com execução inline: `r <code>`
  - Temos uma média de `r mean(mtcars$wt)*1000`lb por carro
- Código em chunk:

```
```{r}
summary(cars)
```
```

- Embarcar gráficos não é problema

```
```{r, echo=FALSE}
plot(cars)
```
```

4 - Lista de exercícios para fixação

- Baixar o pacote de exercícios de:
  - https://dl.dropboxusercontent.com/u/22050262/Lista1-LinguagemR.zip
- Escreva um relatório estruturado na forma de um arquivo Markdown com os resultados abaixo (explique os comandos no arquivo)
  - Importar os dados de ResultadosRTLS.csv
  - Fazer o gráfico de barras (lado a lado) comparando os valores de LM e MCMC-LM, com o título: “Comparação entre os erros resultantes dos métodos LM e MCMC-LM”. Ponha título no eixo x (“experimento”) e no eixo y (“erro (m)”). Use tanto o comando básico quanto `ggplot2`.
  - Mostrar o erro médio das coluna LM e MCMC-LM na forma de texto corrido
  - Mostrar os índices em que a coluna LM é maior do que MCMC-LM (ocultando o comando para tal)
  - Importar COTAHIST.A1997 no formato FWF. Usar as informações do arquivo SeriesHistoricas_Layout.pdf para definir o formato de importação. Limite o número de entradas lidas para 20000.
  - Fazer os gráficos de dispersão entre os valores de abertura de quatro ações (duas a duas) à sua escolha. Serão 6 gráficos a serem mostrados de forma combinada
  - Fazer um gráfico de pizza comparando o volume total negociado dos 7 papéis mais negociados do primeiro dia de pregão, mostrando o volume no gráfico e o código do papel na legenda

Referências

- Norman Matloff. *The Art Of R Programming*. No Starch Press, São Francisco, CA, 2011.
- Joseph Adler. *R in a Nutshell*. O'Reilly Media, Inc., Sebastopol, CA, 2012
- Mark Gardener. Beginning R: The Statistical Programming Language. John Wiley & Sons, Indiana, IN, 2012.
- Robert Kabacoff. *R in Action*. Manning Publications Co., Shelter Island, NY, 2011.
- Winston Chang. R Graphics Cookbook. O'Reilly Media, Inc., Sebastopol, CA, 2013

FIAP

A MELHOR FACULDADE DE TECNOLOGIA