

PG_SERE/INPE

Estatística Aplicada ao Sensoriamento Remoto II

Professor: Thiago S. F. Silva

E-mail: tsfsilva@rc.unesp.br

Exercício 2 – Análise Exploratória e Gráfica

Neste exercício, praticaremos o uso de medidas de tendência central e dispersão, e o uso de ferramentas gráficas.

Os exercícios deverão ser necessariamente realizados utilizando a linguagem R. Para cada exercício, serão dadas algumas dicas, mas “quebrar a cabeça” é parte do processo de aprendizado de qualquer linguagem/software.

Algumas informações úteis:

- O texto em `cinza` indica o código em R. `#` identifica comentários (não são executados).
- O texto de ajuda sobre qualquer função do R pode ser obtido digitando-se “?” seguido do nome da função (ex: `?mean`).
- Este site possui vários exemplos de gráficos feitos em R, com o código equivalente: <http://gallery.r-enthusiasts.com/>
- Google is your friend.

Este exercício deve ser feito individualmente. Favor entregar um script do R, com as respostas em forma de comentário, seguidas das análises relacionadas, via Moodle.

A) Importando e inspecionando os dados no R:

Os dados usados neste exercício fazem parte do conjunto de dados disponibilizado pelo próprio R. A tabela `mtcars` inclui vários dados sobre 32 diferentes modelos de carro, tomados entre 1973 e 1974. Use `?mtcars` para mais informações.

```

# Carrega a tabela de dados mtcars
data(mtcars)
?mtcars

# Inspeccionando a tabela:

str(mtcars)
head(mtcars)

# Notar que os modelos dos carros são incluídos como row names (nomes de
linha), e não como uma variável em si.
names(mtcars)
row.names(mtcars)

# Para usar os nomes dos carros como uma variável, precisamos adicionar
uma coluna. Como nome é uma variável nominal, usamos factor()
mtcars$model <- factor(row.names(mtcars))

# As variáveis transmission e V/S também são fatores:
# am (0 = automático, 1 = manual).
# vs (0 = cilindros em V, 1 = cilindros em linha)
# Então fazemos a correção:

mtcars$am <- factor(mtcars$am)
mtcars$vs <- factor(mtcars$vs)

# Para ficar mais fácil de interpretar, damos nomes aos níveis de am e vs:
levels(mtcars$am)
levels(mtcars$am) <- c("automatic","manual")
levels(mtcars$am)

levels(mtcars$vs)
levels(mtcars$vs) <- c("V","S")
levels(mtcars$vs)

```

PERGUNTA 1: Qual a classe de objeto usada pelo R para armazenar os dados? Quais as características dessa classe, e por que ela é tão importante no R?

Dicas: `class()`

PERGUNTA 2: Quantas observações existem na tabela? E quantas variáveis? Quantas variáveis são numéricas, e quantas são categóricas?

Dicas:

`str ()` mostra o tipo e estrutura de cada variável

`length()` dá o comprimento (quantidade de valores) em um vetor (`vector`)

`dim()` dá as dimensões (linhas e colunas) de um objeto 2D, como uma `matrix` ou `data.frame`. Objetos unidimensionais (como vetores) tem dimensão nula.

```
# Exemplos de length() e dim()

#criamos um vetor chamado x. c() concatena valores e cria um vetor
x <- c(1,2,4,8,10)

length(x)

# dim() não funciona pra dados unidimensionais
dim(x)

dim(mtcars)

# Veja que cada coluna isolada de uma data.frame é tratada como um vetor
# Para acessar uma coluna pelo nome, use o nome da data.frame, seguido de $

dim(mtcars$cyl)

length(mtcars$cyl)
```

B) Análise Exploratória

PERGUNTA 3: Quais os valores médios, mínimos e máximos de cada variável numérica na tabela? Quantos níveis existem para cada variável categórica?

Dica: `summary()`

PERGUNTA 4: Qual desvio padrão para cada variável numérica na tabela?

Dicas: `[]`, `apply()`, `sd()`

Colunas e linhas específicas de uma `data.frame` podem ser especificadas usando `[]`:

```
# Exemplo de seleção de linhas e colunas:
```

```
# todas as linhas, colunas 1 a 3
mtcars[,1:3]

# todas as linhas, colunas 1,3 e 5
mtcars[,c(1,3,5)]

# linhas 1 a 5, todas as colunas
mtcars[1:5,]

# linhas 1 a 5, todas as colunas menos as colunas 1 e 2
mtcars[1:5,-c(1,2)]
```

Uma função qualquer pode ser aplicada para cada coluna de uma `data.frame`, usando o comando `apply(x, margin, fun,...)`. `x` é a `data.frame`, `margin` é um valor numérico dizendo se a fórmula deve ser aplicada por linhas (1) ou colunas (2), e `fun` é o nome da função, seguido de quaisquer parâmetros dessa função.

IMPORTANTE: ao usar `apply()`, se a função a ser aplicada não for apropriada para uma dos colunas (ex. média de fatores), os resultados dão erro para todas as colunas:

```
# As colunas "model", "vs" e "am" são fatores, bagunçando toda a análise:
apply(mtcars,2,mean)

# Se excluirmos as colunas não-numéricas, funciona:
apply(mtcars[, -c(8,9,12)],2,mean)

#comparar com as médias de summary()
```

PERGUNTA 5: Dentre todas as variáveis numéricas medidas, quais foram as mais e menos variáveis?

Dica: Podemos criar novas funções no R, usando `function()`. Os parâmetros especificados dentro dos parênteses vão ser os parâmetros que a nova função irá precisar, e o que vem após os parênteses é o cálculo executado pela função. Se a função precisar de mais de uma linha de comandos, podemos usar `{ }`:

```
# Criando novas funções:
# Coeficiente de variação
cv <- function(x) sd(x)/mean(x) * 100

# Mesma coisa, mas usando linhas separadas para cada um dos calculos:
# Nesse caso, precisamos especificar qual dos calculos é o resultado
final, usando return()
```

```

cv2 <- function(x){
  sdval <- sd(x)
  mnval <- mean(x)
  razao <- sdval/mnval
  cvval <- razao * 100
  return(cvval)
}

teste <- c(1,3,5,6,8,9,0)
cv(teste)
cv2(teste)

```

PERGUNTA 6: Calcule os quartis de cada variável numérica, incluindo a mediana. Comparando-se a média e a mediana de cada variável, quais dessas variáveis tem uma distribuição assimétrica?

Dicas: `median()`, `quantile()`

C) Análise gráfica

PERGUNTA 7: Através de análise gráfica, determine qual das três variáveis a seguir menos se aproxima de uma distribuição normal: disp, mpg ou qsec?

Dicas: `hist()` e `plot(density())`

Não esqueça de ajustar os parâmetros `breaks` (para `hist()`) e `bw` (para `density()`)

PERGUNTA 8: Através de um gráfico de barras, mostre a distribuição do número de cilindros e do número de carburadores para os carros amostrados

Dicas: `table()` e `barplot()`

PERGUNTA 9: Através de análise gráfica, mostre como o tempo mínimo que cada carro leva para percorrer 1/4 de milha (qsec) e o consumo (mpg) são relacionados à potencia (hp) do motor. Nomeie os eixos de cada gráfico adequadamente.

Dicas: `plot (x, y, xlab="Nome eixo X", ylab="Nome eixo Y")` é equivalente a `plot (y ~ x, data = nome data.frame, xlab="Nome eixo X", ylab="Nome eixo Y")`

PERGUNTA 10: Através de *boxplots*, avalie as hipóteses de que a potência do motor (hp) está relacionada com o a) número de cilindros (cyl) e b) com a quantidade marchas (gear). Reporte suas conclusões, além dos gráficos.

Dicas: `boxplot (y ~ x, data = nome data.frame)`

PERGUNTA 11: Usando a sintaxe ggplot2 abaixo, discuta a influência do peso (wt) de cada carro sobre a relação potência (hp) x tempo para percorrer 1/4 de milha (qsec).

```
library(ggplot2) # instalar o pacote ggplot2 se não estiver instalado

ggplot(mtcars,aes(hp,qsec)) + geom_point(aes(color=wt),size=5) +
xlab("Potência (hp)") + ylab("Tempo em 1/4 de milha (seg)") +
scale_color_continuous(name="Peso\n(libras/1000)")
# No futuro, espero dar uma aula extra só sobre ggplot2. Por enquanto,
confiem no que eu escrevi ☺. Quem tiver curiosidade pode fuçar em
http://ggplot2.org/
```

PERGUNTA 12: Calcule uma nova variável para a relação peso/potência (pp), e faça um *scatterplot* destes valores contra qsec. Explique o efeito desta transformação sob a direção e a força da correlação com qsec, em comparação com o plot de hp x qsec da pergunta 8.

Dicas: `data.frame$varnova <- data.frame$var1 / data.frame$var2`

PERGUNTA 13: No plot anterior, um dos carros aparenta ser um *outlier*. Que carro é esse?