



# **Curso de introdução ao software R - Modulo 1**

Darlene Duarte

Renata Veroneze

# Gráficos com estilo - ggplot2



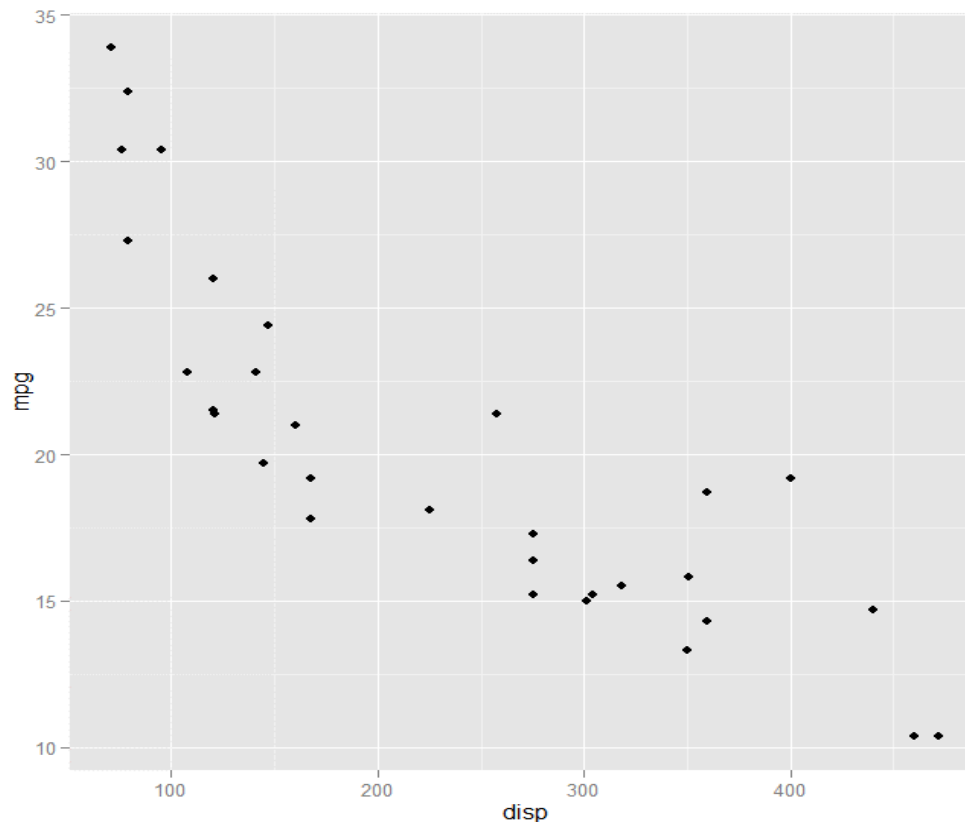
- Qual o primeiro passo para utilizar as funções de um pacote?
- Vamos usar o banco de dados do pacote

```
> head(mtcars)
```



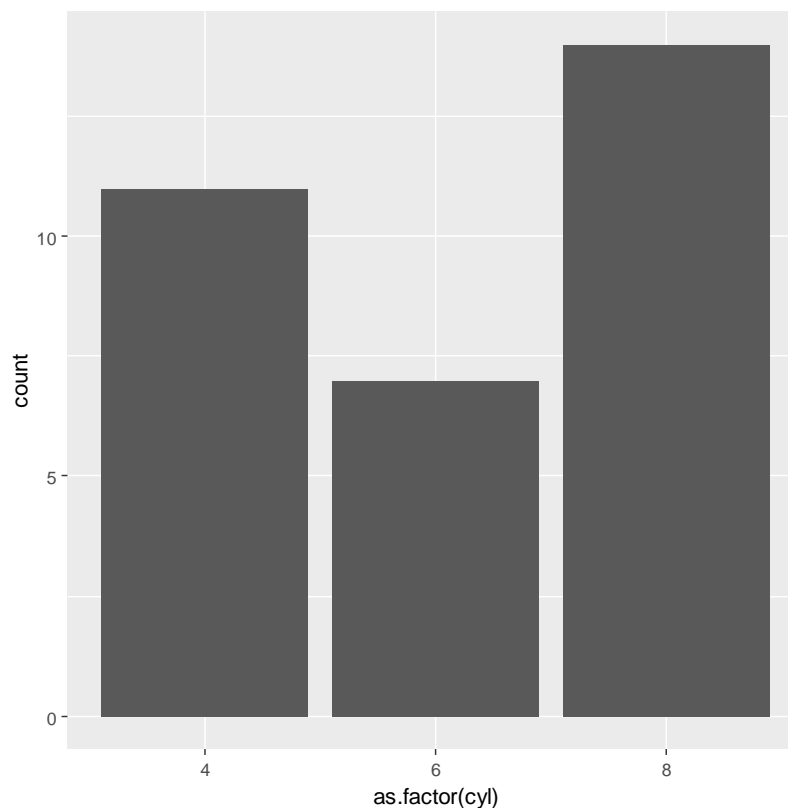
# ggplot2 – Gráfico de pontos

```
ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point()
```



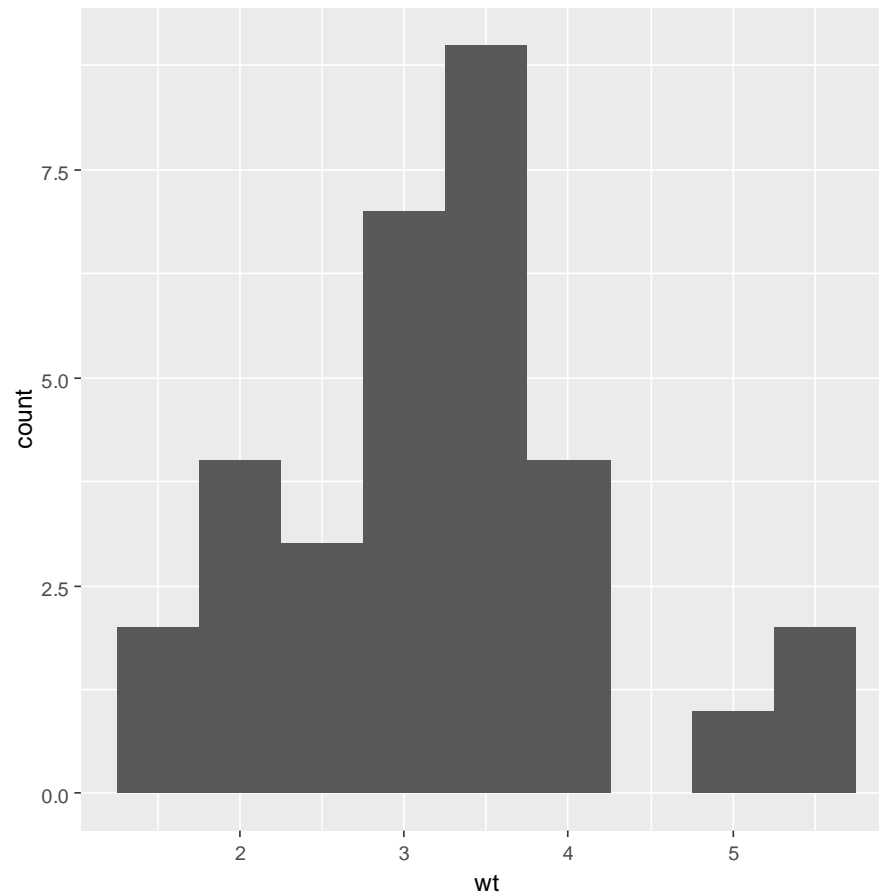
# ggplot2 – Gráfico de barras

```
ggplot(mtcars, aes(x = as.factor(cyl))) +  
  geom_bar()
```



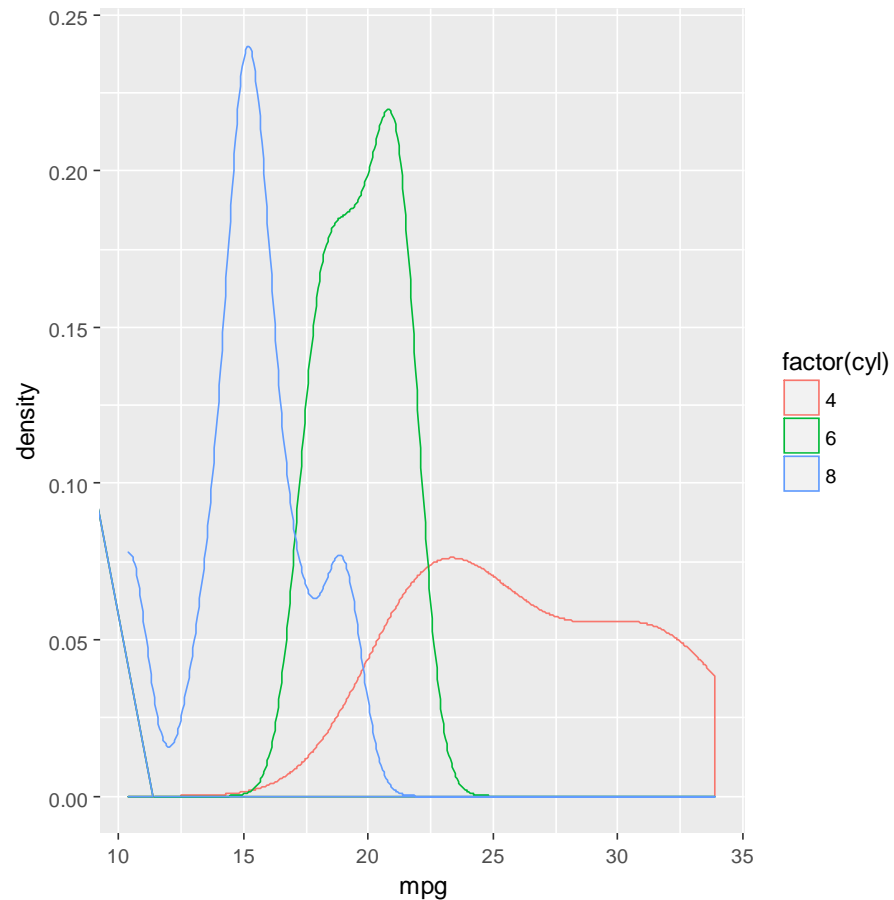
# ggplot2 – Histograma

```
ggplot(mtcars, aes(x = wt )) +  
  geom_histogram(binwidth=.5)
```



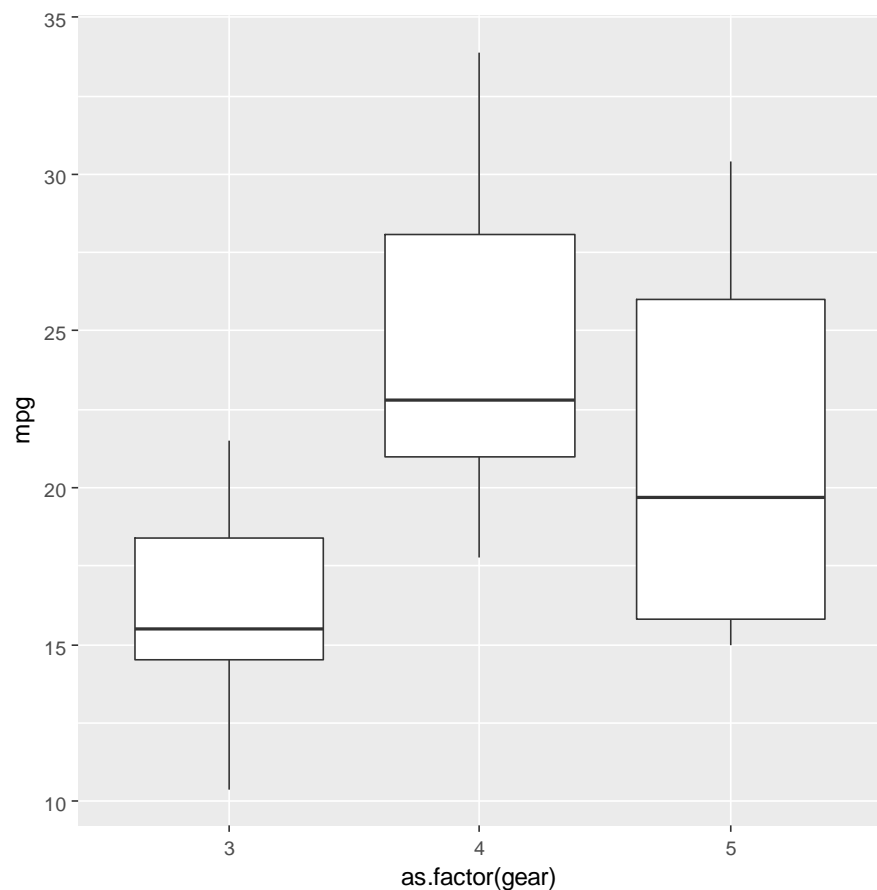
# ggplot2 – Histograma

```
ggplot(mtcars, aes(x=mpg, colour=factor(cyl))) +  
  geom_density()
```



# ggplot2 – boxplot

```
ggplot(mtcars, aes(x = as.factor(gear), y = mpg)) +  
  geom_boxplot()
```



# Desafio



1. Utilizando o ggplot2 e o exercicio3.5\_2.txt
2. Faça um plot utilizando as variáveis dap e ht
3. Faça um boxplot utilizando as variáveis dap e classe
4. Faça um histograma utilizando a variável folha





# Resposta



```
a<-read.table("exercicio3.5_2.txt",h=T, sep=",")
```

```
head(a)
```

```
>ggplot(data =a, aes(x = dap, y = ht)) + geom_point()
```

```
>ggplot(data=a, aes(x =classe, y =dap)) +  
  geom_boxplot()
```

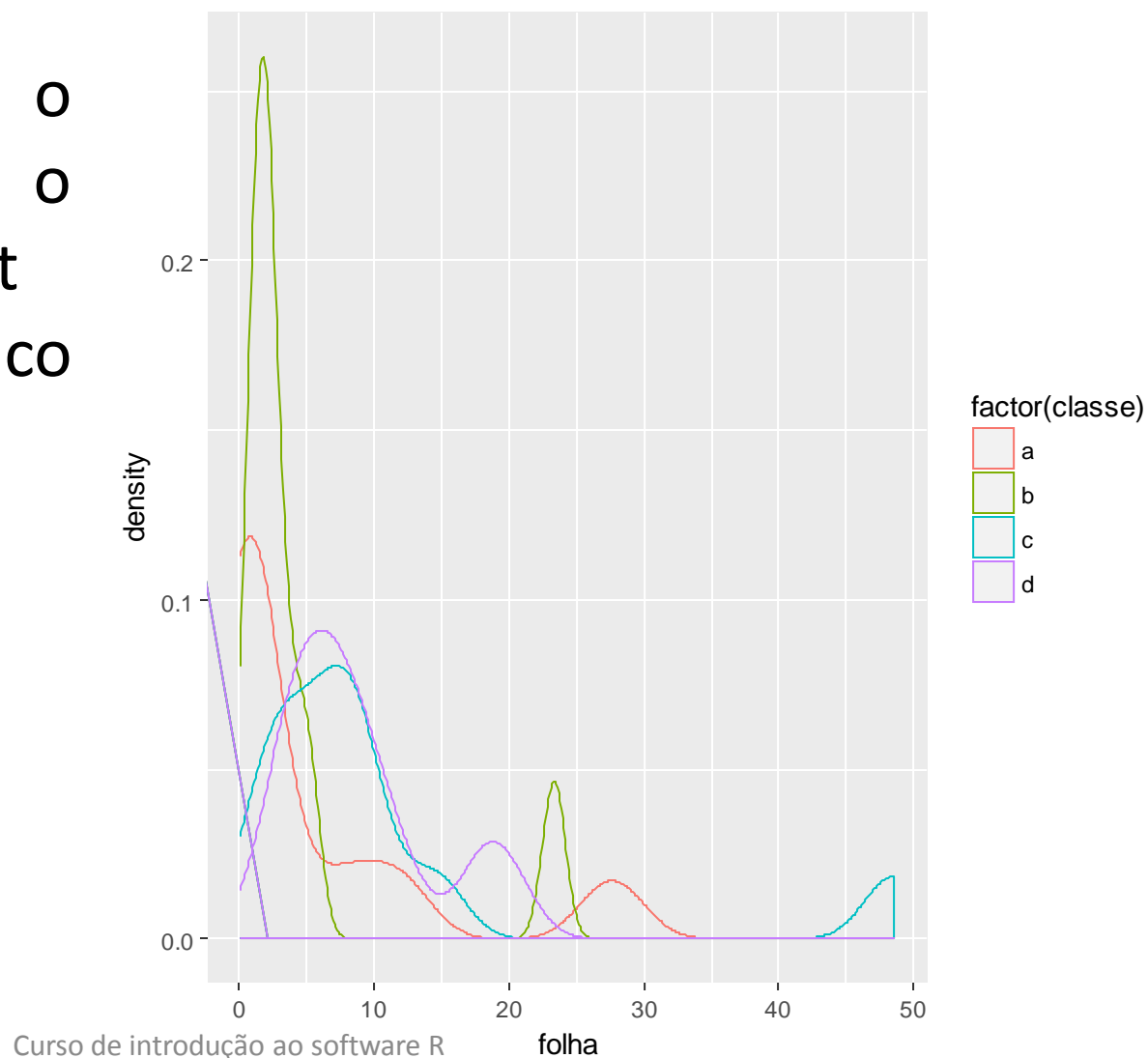
```
>ggplot(data=a, aes(x =folha )) +  
  geom_histogram(binwidth=1)
```



# Desafio



1. Utilizando o `ggplot2` e o `exercicio3.5_2.txt` reproduza o gráfico ao lado



# Resposta

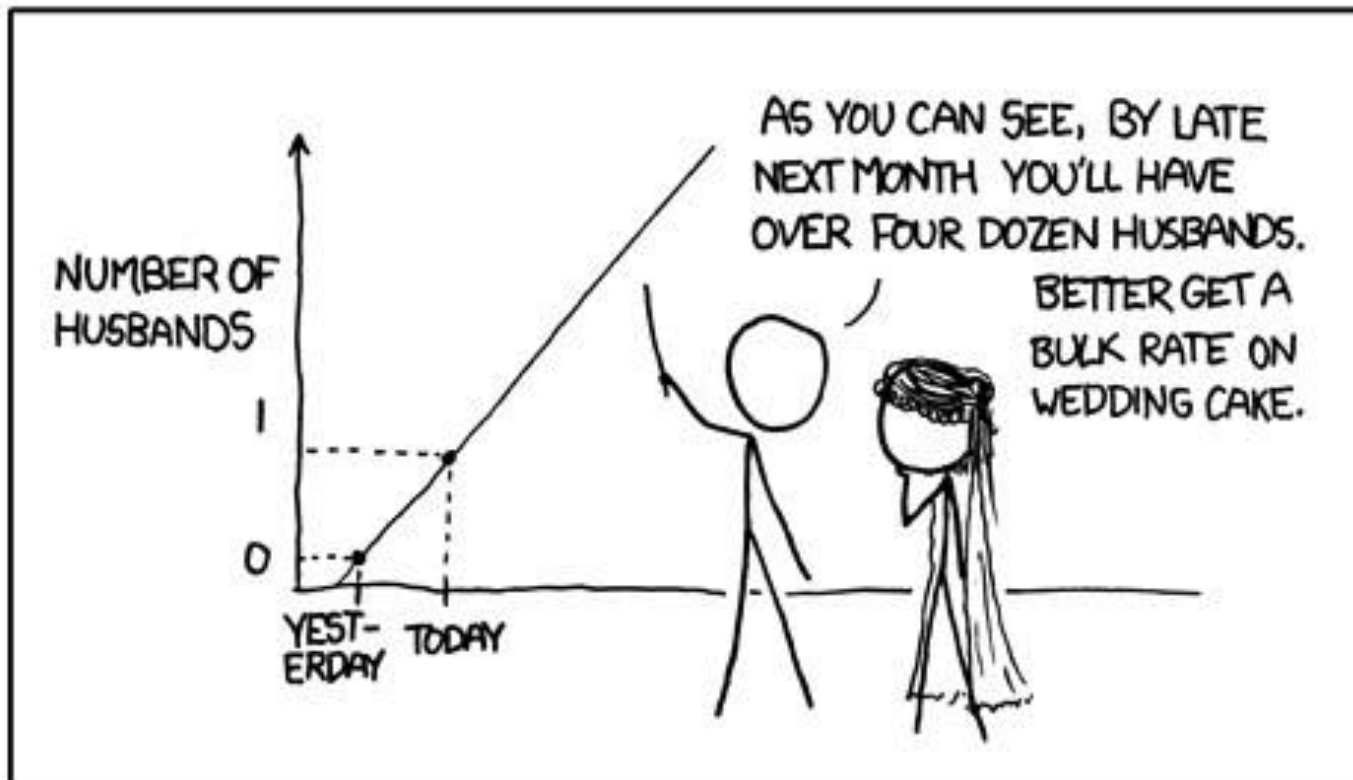


```
ggplot(data=a, aes(x=folha, colour=factor(classe))) +  
  geom_density()
```



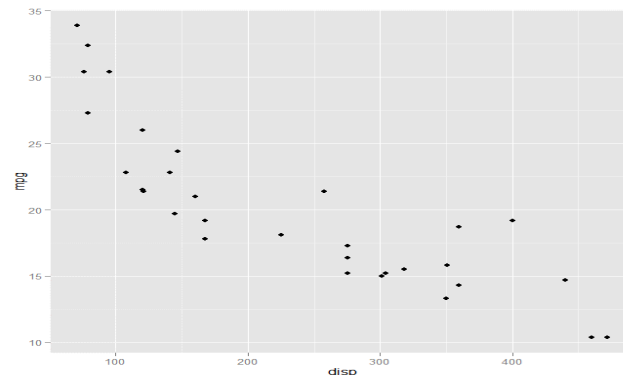
# Gráficos

## MY HOBBY: EXTRAPOLATING



# Editando e salvando gráficos

Mudando a cor dos pontos



```
> ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point(colour="red")
```

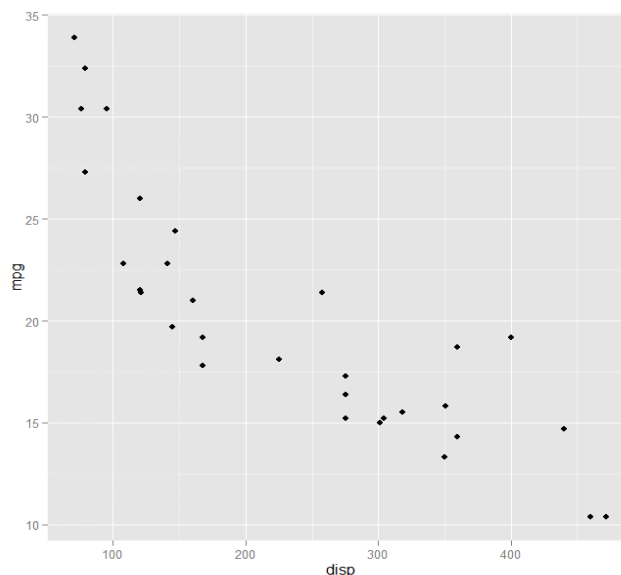
```
> ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point(colour="blue")
```



# Editando e salvando gráficos

Alterar o tamanho dos pontos

```
> ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point(size=4)
```



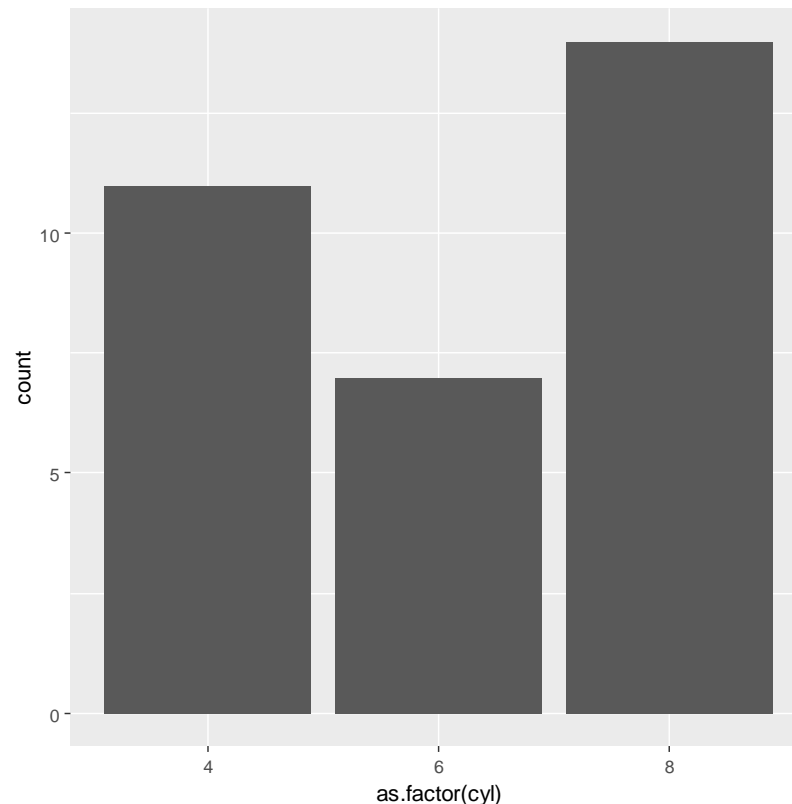
# Acrescentar outra variável

```
> ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point(aes(colour = cyl)) +  
  scale_colour_gradient(low = "red")
```

```
> ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
  geom_point(colour="blue", aes(shape =  
  factor(cyl)))
```

# Mudando as opções de cores do gráfico de barras

```
> ggplot(mtcars, aes(x = as.factor(cyl))) +  
  geom_bar(fill = "white", colour = "red")
```





# Mudando as opções de cores do gráfico de barras

```
> ggplot(mtcars, aes(x = as.factor(cyl))) +  
geom_bar(aes(fill = factor(cyl)))#barras de cores  
diferentes
```

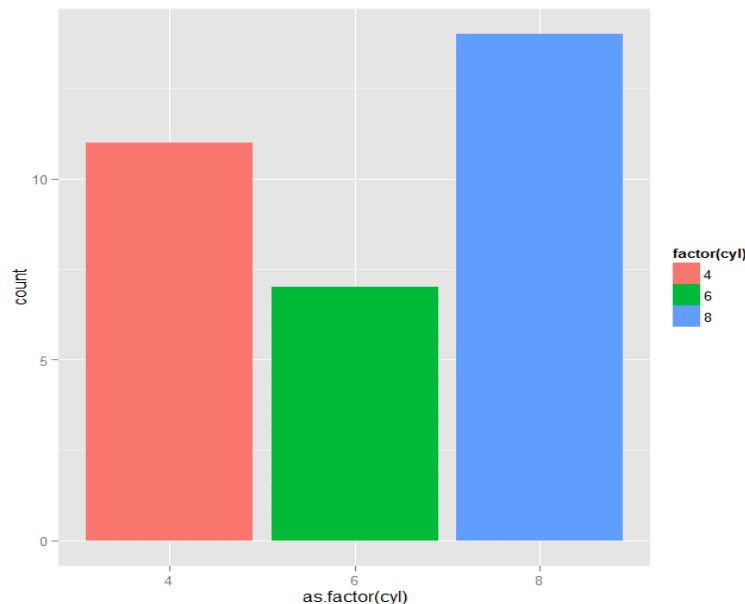
```
> ggplot(mtcars, aes(x = as.factor(cyl), y = mpg,  
colour = as.factor(cyl))) + geom_boxplot()
```

```
> ggplot(mtcars, aes(x = as.factor(cyl), y = mpg)) +  
geom_boxplot(fill=c("orange","yellow","green"))
```

# Legendas

Alterar o padrão das legendas

```
>ggplot(mtcars, aes(x = as.factor(cyl), fill =  
as.factor(cyl))) + geom_bar() + labs(fill = "cyl") +  
theme(legend.position="left")
```



# Título

Colocar título

```
>ggplot(mtcars, aes(x = as.factor(cyl), fill =  
as.factor(cyl))) + geom_bar() + labs(title = " New  
plot title") + labs(fill = "cyl") +  
theme(legend.position="left")
```



# Título

Mudando as opções do título:

```
> ggplot(mtcars, aes(x = as.factor(cyl), fill =  
as.factor(cyl))) + geom_bar() + labs(fill = "cyl") +  
labs(title = " New plot title") +  
theme (plot.title = element_text(family = "sans",  
face = "italic", colour = "red", size = 12))+  
theme(plot.title = element_text(hjust = 0.5))
```



# Anotação no gráfico

Anotação no gráfico

```
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() +  
annotate("text", x = 4.5, y = 30, label = "Texto")
```



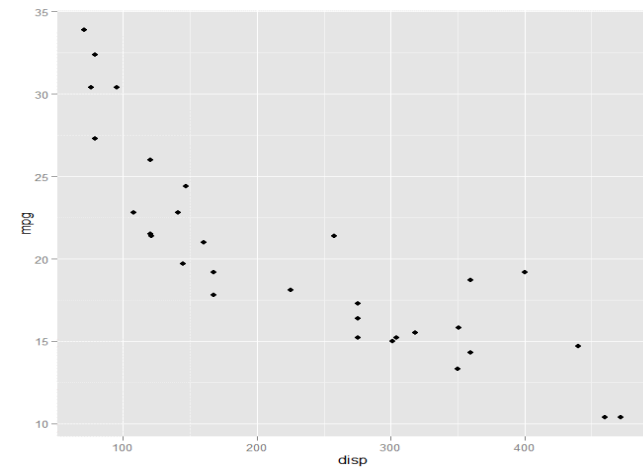
# Eixos

Mudando os nomes dos eixos

```
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() +  
xlab("eixo X") + ylab("eixo Y") +  
theme(axis.title=element_text(size=14,face="bold"))
```



# Fundo do gráfico

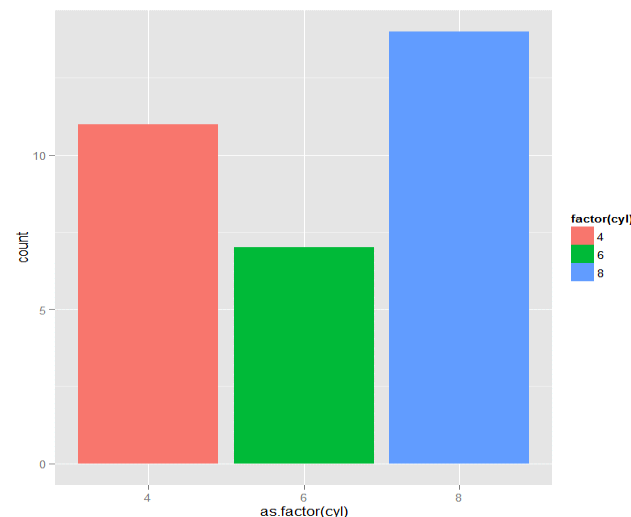


```
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() +  
theme_bw()
```

```
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() + theme(panel.grid =  
element_blank())
```

# Fundo do gráfico

```
ggplot(mtcars, aes(x = as.factor(cyl), fill = as.factor(cyl))) +  
geom_bar() +  
theme(  
  plot.background = element_blank()  
  ,panel.grid.major = element_blank()  
  ,panel.grid.minor = element_blank()  
  ,panel.background = element_blank()  
  
  ) + theme(axis.line = element_line(color = 'black'))
```





# Salvando gráficos

Existem vários formatos em que o R pode salvar imagens gráficas: JPEG, BMP, PDF, TIFF, PNG.

Abaixo há um exemplo utilizando o formato BMP, mas o mesmo pode ser feito para qualquer formato.



# Salvando gráficos

```
> bmp(file="grafico.bmp")  
> ggplot(mtcars, aes(x = wt, y = mpg)) +  
  geom_point() +  
  theme_bw()  
> dev.off()
```



# Janelas

```
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() +  
annotate("text", x = 4.5, y = 30, label = "Texto")  
>windows()  
>ggplot(mtcars, aes(x = wt, y = mpg)) +  
geom_point() +  
xlab("eixo X") + ylab("eixo Y") +  
theme(axis.title=element_text(size=14,face="bold")  
)
```



# Exercício

Usando os dados do arquivo `exercicio3.5`, construa um gráfico de pontos considerando o diâmetro na altura do peito (`dap`) e a altura (`ht`), nos eixos. Adicione também informação sobre a região.



# Resposta

```
>ggplot(b, aes(x = dap, y = ht)) +  
geom_point(aes(colour=regiao))
```

# Exercício

Coloque o seguinte título “Diâmetro x Altura”, usando a fonte “serif”, com tamanho 14, em itálico e na cor vermelha.



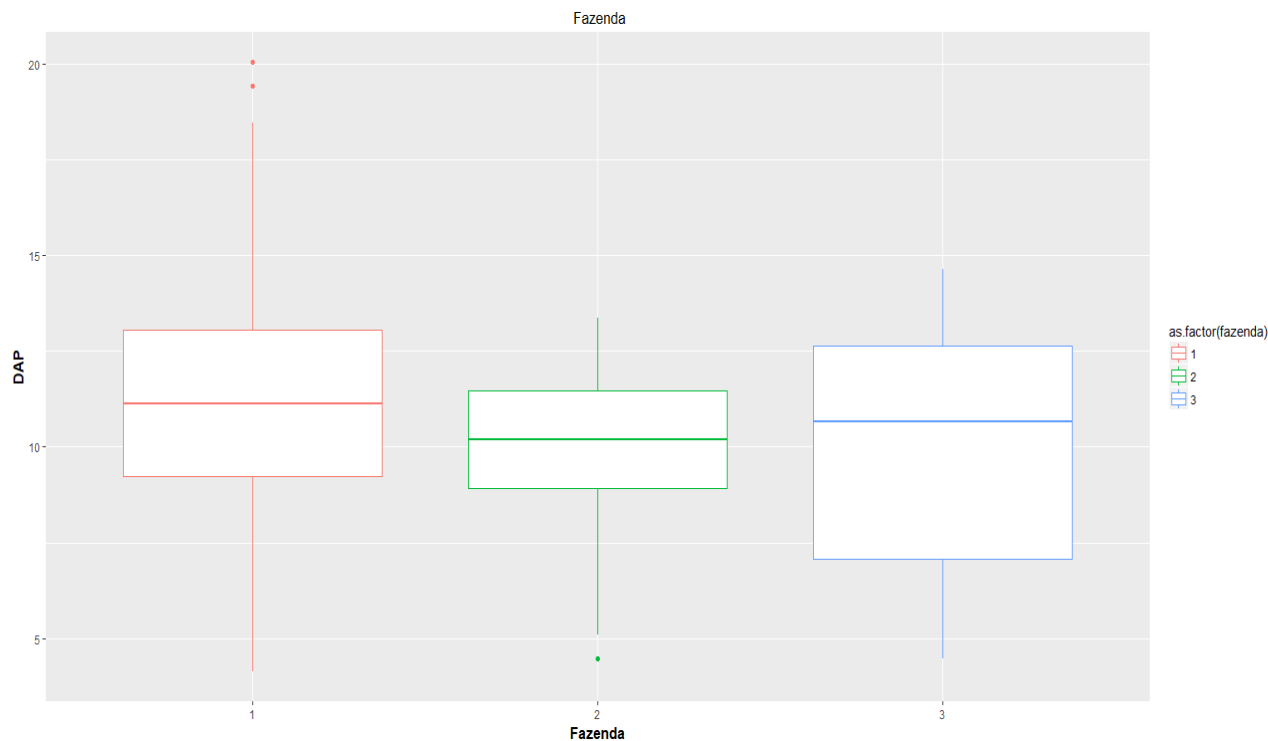
# Resposta

```
>ggplot(b, aes(x = dap, y = ht)) +  
geom_point(aes(colour=regiao)) +  
labs(fill = "cyl") + labs(title = " Diâmetro X  
Altura") +  
theme (plot.title = element_text(family = "serif",  
face = "italic", size = 14))
```



# Exercício

Considerando o diâmetro na altura do peito (dap) das três primeiras fazendas. Obtenha o gráfico a baixo.





# Resposta

```
>novo<-subset(b, b$fazenda <= 3)
```

```
>ggplot(novo, aes(x = as.factor(fazenda), y = dap,  
colour = as.factor(fazenda))) + geom_boxplot() +  
xlab("Fazenda") + ylab("DAP") + labs(title = "  
Fazenda") +  
theme(axis.title=element_text(size=12,face="bold")  
) + theme(plot.title = element_text(hjust = 0.5))
```



# ANOVA

DIC

```
>diam<-read.table("exemplo_dic.txt",h=T)
```

```
>head(diam)
```

```
>a0 <- aov(diam~gen, data=diam)
```

```
>summary(a0)
```



# ANOVA

```
>shapiro.test(residuals(a0))  
>bartlett.test(residuals(a0)~diam$gen)  
>?bartlett.test  
>?shapiro.test  
>TukeyHSD(a0)
```



# ANOVA

DIC em esquema factorial

```
>vol <- read.table("dic_fatoria.txt", h=T)
```

```
>head(vol)
```

```
>table(vol$gen)
```

```
>table(vol$dose)
```

```
>class(vol$gen)
```

```
>class(vol$dose)
```

```
>vol$dose <- factor(vol$dose)
```

```
>class(vol$dose)
```



# ANOVA

```
>m0 <- aov(volu~gen+dose+gen:dose, data=vol)
```

```
>summary(m0)
```

```
>m0 <- aov(volu~gen*dose, data=vol)
```

```
>summary(m0)
```

```
>TukeyHSD(m0)
```



# ANOVA

DBC

```
>mad<-read.table("madeira.txt",h=T)
>head(mad)
>m0 <- aov(prod~bloco+proced, data=mad)
>summary(m0)
#teste de comparação de médias
TukeyHSD(m0)
```



# Exercício

Faça uma anova (DIC) utilizando o arquivo dados\_suinos.csv considerando a característica CMD.



# Resposta

```
> ex<-read.table("dados_suinos.csv",h=T,sep=",")  
>head(ex)
```

```
>table(ex$TRAT)
```

```
>ex$TRAT <- factor(ex$TRAT)
```

```
>an <- aov(CMD~TRAT, data=ex)
```

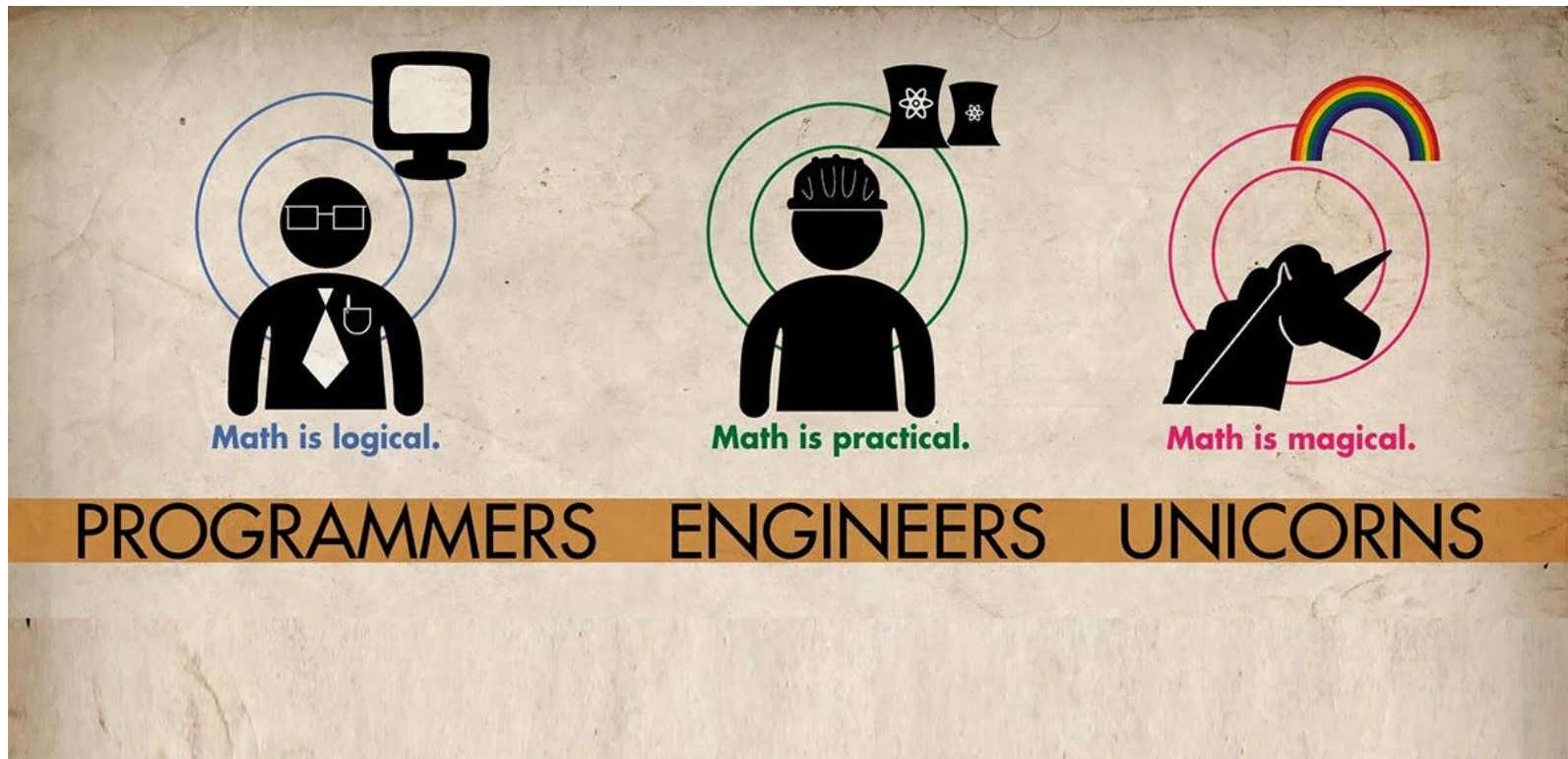
```
>summary(an)
```

```
>TukeyHSD(an)
```





# Funções e programação



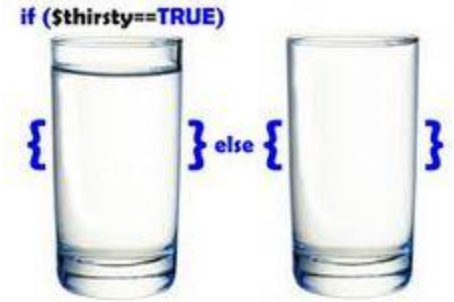
# IF

- Ordem condicionais

```
>x<-6
```

```
>y<-4
```

```
>if(x == 6 & y>3) print("Verdadeiro") else  
  print("Falso")
```

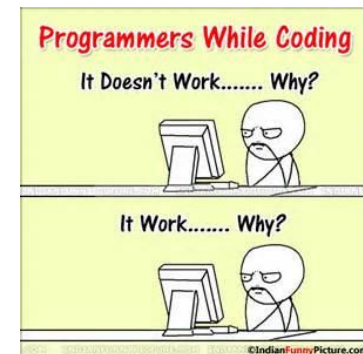


# IFELSE

- Vetores: ifelse
- ifelse(condição, expres\_1 , expres\_2)

```
>salarios<-c(1000, 400, 1200, 3500, 380, 3000, 855,  
             700, 1500, 500)
```

```
> ifelse(salarios<1000,"pouco","muito")
```



# FOR



- Loop: execução de uma série de comandos repetidas vezes
- `for(i in 1:n){comandos}`

```
resu<-numeric(0)#criando objeto vazio
```

```
>for(i in 1:5)
{
  resu[i]<-i^2
}
```



# FOR



- Fibonacci é uma sequência famosa na matemática
- Os dois primeiros números da sequência são [1, 1]. Os números subsequentes são compostos pela soma dos dois números anteriores



# FOR

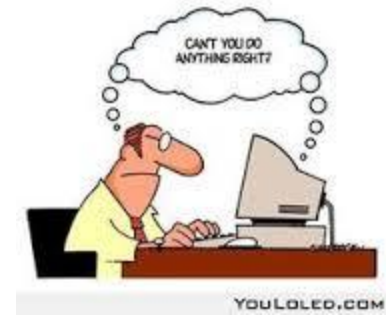


- Vamos usar a função for para descobrir os 12 primeiros números da seqüência de Fibonacci

```
Fibonacci<-numeric(0)
Fibonacci[c(1,2)]<-1 # o 1º e 2º valores da seqüência
devem ser 1
for (i in 3:12) # 3 a 12 porque já temos os dois
primeiros
{
  Fibonacci[i]<-Fibonacci[i-2]+Fibonacci[i-1]
}
Fibonacci
```



# FUNCTION



- Podemos criar nossas funções!!
- `nome_da_funcao<-function(argumento1, argumento2, ..., argumento n)`



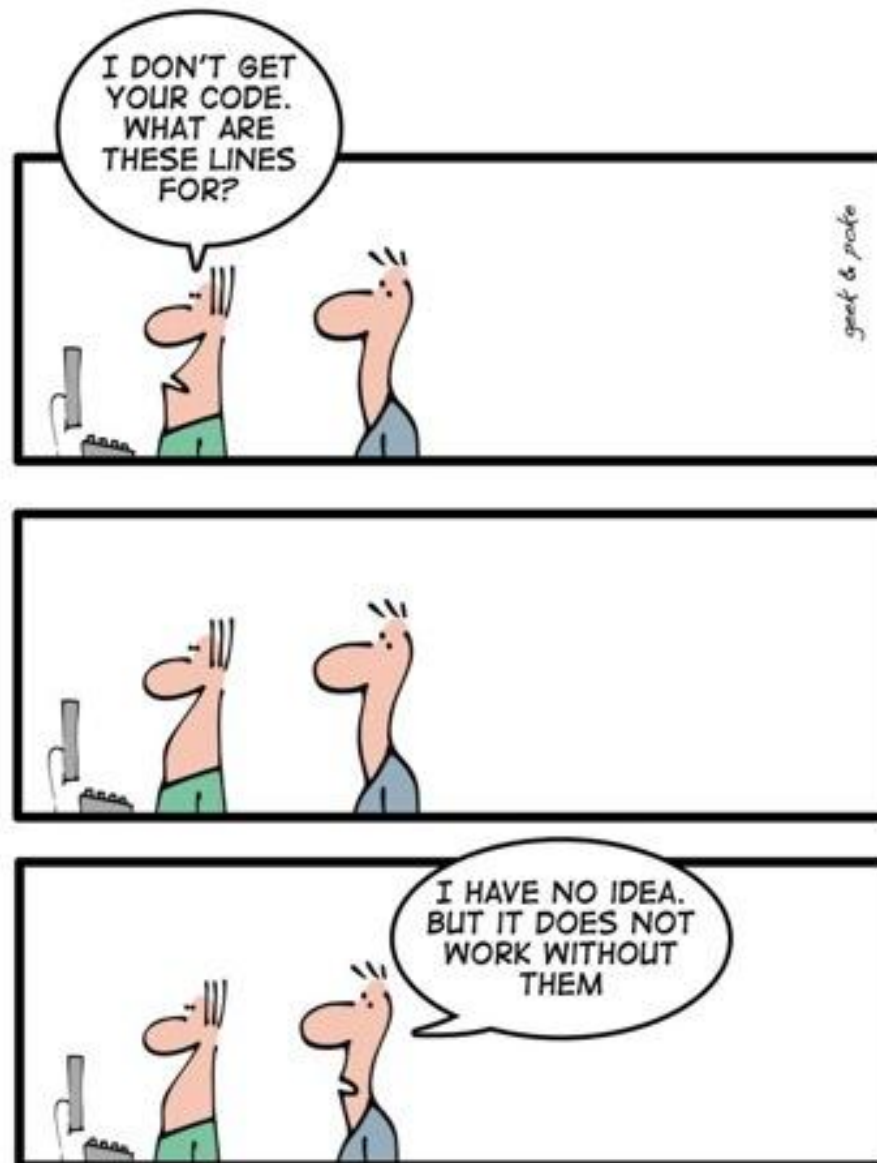
# FUNCTION

```
media<-function(dados)
{
  print(sum(dados)/length(dados))
}
```





# Funções e programação



THE ART OF PROGRAMMING - PART 2: KISS



# Desafio



1. Leia o arquivo `simu.txt`. Este arquivo contém amostras de Simuliidae (borrachudos - piuns) coletadas em 50 riachos da Chapada Diamantina - Bahia. Note que as 6 primeiras variáveis são ambientais e depois os dados de 20 espécies. Estes dados foram coletados em três municípios dentro da chapada diamantina (Lençóis, Mucugê e Rio de Contas). Crie um objeto `spp` contendo somente os dados das espécies (colunas 7 a 26) e transforme os dados em presença e ausência. Dica: crie uma condicional de modo que se o valor de `spp` for maior ou igual a 1 seja 1, se não, seja 0.



# Resposta



```
setwd("C:\\Users\\Renata\\Documents\\curso_R_2017  
\\curso_R")
```

```
simuli<-read.table("simu.txt",h=T)
```

```
head(simuli)
```

```
spp<-simuli[,7:26]
```

```
head(spp)
```



# Desafio



2. Construa uma função chamada "**conversor**" que calcula automaticamente o valor de graus Celsius, sabendo-se a temperatura em Fahrenheit. Esta função deve retornar um data frame com duas colunas, uma contendo os valores em Celsius e a outra contendo os valores em Fahrenheit. A função matemática que descreve essa conversão é:

$$C = \frac{5}{9}(F - 32)$$



# Resposta



```
ifelse(spp>=1,1,0)  
head(spp)
```

```
dado <- data.frame()  
conversor <- function(dado)  
{  
  C=(5/9)*(dado-32)  
  resultado <- data.frame(dado,C)  
  return(resultado)  
}
```

```
data<-c(0,10,20,50,100)
```

```
conversor(data)
```

