

Análise de imagem no Rstudio

Ana, João, Laura, Leonardo e Paulo

20 de novembro de 2019

Formatos de imagens

- TIFF
- JPEG
- PNG
- SVG
- GIF
- BMP
- PDF
- EPS

A importância de análise de imagens

Pacotes

Oos principais pacotes para manipulação de imagem são:

```
require("BiocManager")  
require("EBImage") # JPEG(JPG), PNG E TIFF  
  
require("imager") # JPEG(JPG), PNG E BMP  
  
require("magick")
```

Importação e visualização de imagens:

- EImage:

```
.ima <- readImage("C:/Users/nick_/Downloads/897207.jpg")  
.display(ima)
```

- Imager:

```
.ima_1 <- load.image("C:/Users/nick_/Downloads/897207.jpg")  
.plot(ima_1)
```

- Magick:

```
.ima_2 <- image_read("C:/Users/nick_/Downloads/897207.jpg")  
.print(ima_2)
```

Mudar dimensões

```
tigre <- image_read_svg('http://jeroen.github.io/images/tiger.svg')  
tigre
```



Mudar dimensões

```
tigre2 <- image_read_svg('http://jeroen.github.io/images/tiger.svg',  
                          width = 120) # 120 = width, 120x = height  
tigre2
```



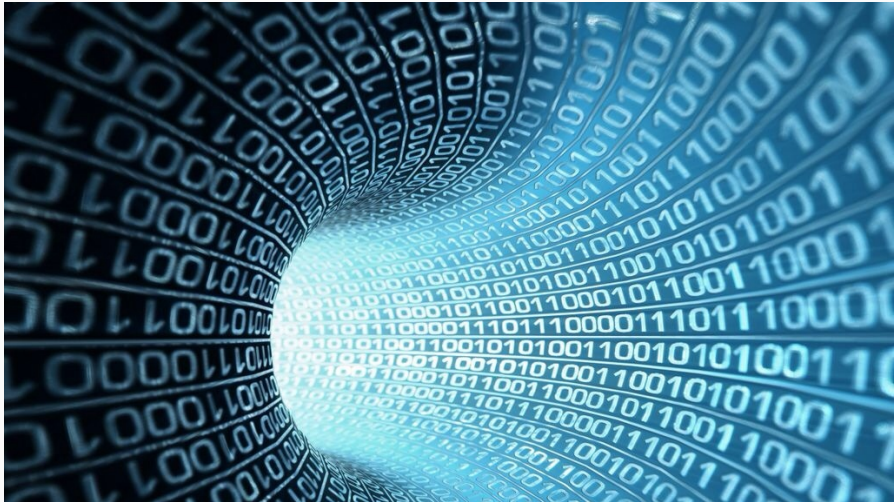
```
tigre_redimensionada <- image_scale(tigre, "120x120")
```

Converter ou salvar em formatos desejados

```
tigre_convertido <- image_convert(tigre, "jpeg")  
image_info(tigre_convertido) # Retorna o formato da imagem  
  
##    format width height colorspace matte filesize density  
## 1    JPEG   900    900         sRGB  TRUE         0    72x72  
  
image_write(tigre, path = "tiger.png", format = "png")
```


Imagens para manipulação (BigData)

```
bigdata <- image_read('C:/Users/nick_/OneDrive/Área de Trabalho/BigData/BigData.png')  
bigdata
```



Imagens para manipulação (R)

```
logo <- image_read('C:/Users/nick_/OneDrive/Área de Trabalho  
logo
```



Girar e modificar

```
image_flop(frink)
```



```
image_flip(frink)
```



```
image_rotate(frink, 45)
```



```
image_crop(frink, "100x150+50")
```



Alguns tipos de filtros

```
image_blur(frink, 10, 5)
```



```
image_charcoal(frink)
```



```
image_oilpaint(frink)
```



```
image_negate(frink)
```



```
image_modulate(frink, brightness = 80, saturation = 120, hue = 90)
```



```
image_fill(frink, "orange", point = "+100+200", fuzz = 20)
```



Imagens sobrepostas

```
img <- c(bigdata, logo, frink)
img <- image_scale(img, "300x300")
image_info(img)
```

##	format	width	height	colorspace	matte	filesize	density
## 1	JPEG	300	225	sRGB	FALSE	0	72x72
## 2	PNG	300	232	sRGB	TRUE	0	72x72
## 3	PNG	148	300	sRGB	TRUE	0	72x72

Imagens sobrepostas

```
image_append(image_scale(img, "100"), stack = TRUE)
```



```
image_append(image_scale(img, "x200"))
```



```
image_mosaic(img)
```



```
image_flatten(img, 'Minus')
```



```
image_flatten(img)
```

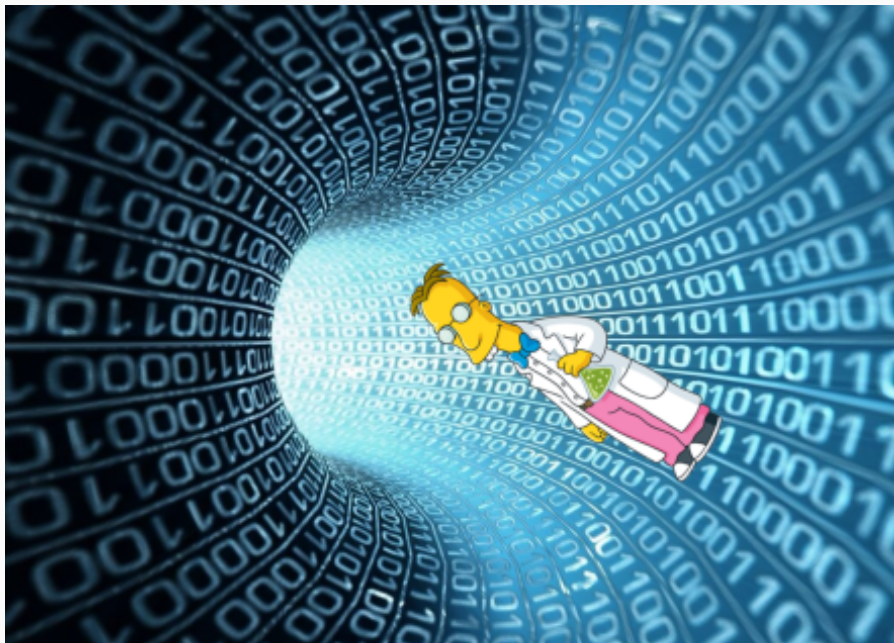


Fonte: <https://cran.r-project.org/web/packages/magick/vignettes/intro.html>

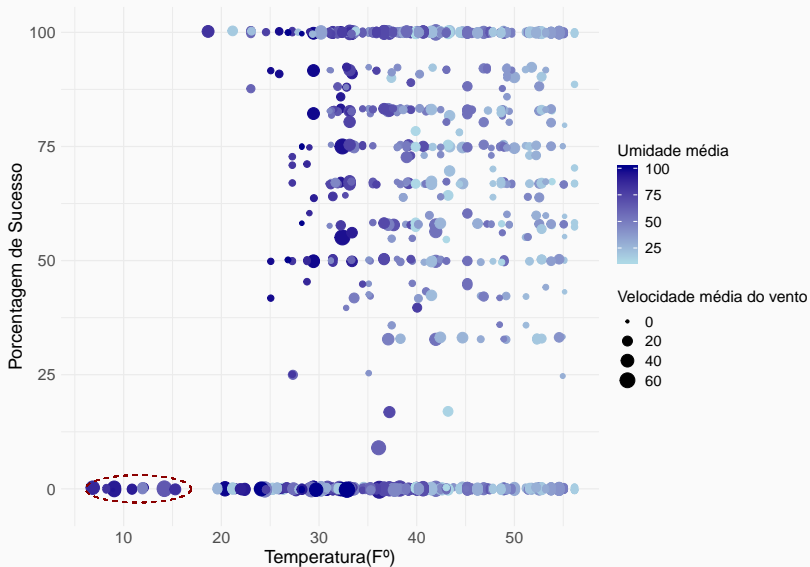
Imagens sobrepostas

```
bigdatafrink <- image_scale(image_rotate(  
  image_background(frink, "none"), 300), "x160")  
juntos <- image_composite(image_scale(  
  bigdata, "x330"), bigdatafrink, offset = "+180+100")
```


Imagens sobrepostas



Utilidade em gráficos



Utilidade para sobreposição de imagens

```
graph <- image_read("C:/Users/nick_/OneDrive/Área de Trabalho/Temp.png")
temp <- image_read("C:/Users/nick_/OneDrive/Área de Trabalho/Graph.png")
temp_graph <- image_scale(image_rotate(image_background(
  temp, "none"), 340), "x50")
temp_graph
```



```
juntos_2 <- image_composite(image_scale(
  graph, "x600"), temp_graph, offset = "+150+440")
image_write(juntos_2, path = "juntos2.pdf", format = "pdf")
```

Utilidade para sobreposição de imagens



Anotações em imagens

```
frink_anot <- image_annotate(frink, "Aqui", size = 25,  
                             color = "red",  
                             boxcolor = "black",  
                             degrees = 30,  
                             location = "+150+310")  
  
frink_anot <- image_scale(frink_anot, "x350")  
  
image_write(frink_anot, path = "frink_anot.png", format = "png")
```

Anotações em imagens



```
earth <- image_read("https://jeroen.github.io/images/earth.  
  image_scale("250x") %>%  
  image_quantize()  
  
length(earth)
```

```
## [1] 44
```

Como montar um GIF

1º - Importe as imagens:

```
⇒ im_1 <- image_read("C:/Users/nick_/Downloads/im_1.jpg")
```

```
⇒ im_n <- image_read("C:/Users/nick_/Downloads/im_n.jpg")
```

2º - Junte as imagens e redimensione:

```
⇒ img <- c(im_1, ... , im_n)
```

```
⇒ img <- image_scale(img, "300x300")
```

3º - Argumentos:

```
⇒ image_animate(img)
```


Salvar o GIF na máquina

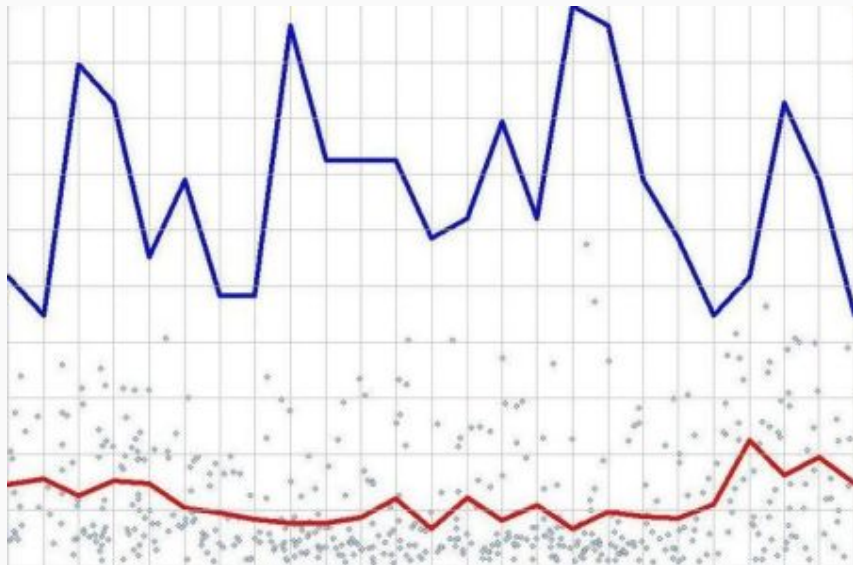
```
library(gifski)

#image_write_gif(img, path = "grafico.gif")

# É possível adicionar um delay

#image_write_gif(img, path = "grafico_delay.gif",
#                delay = 1/6)
```

Como retirar pontos de um gráfico sem seu código?



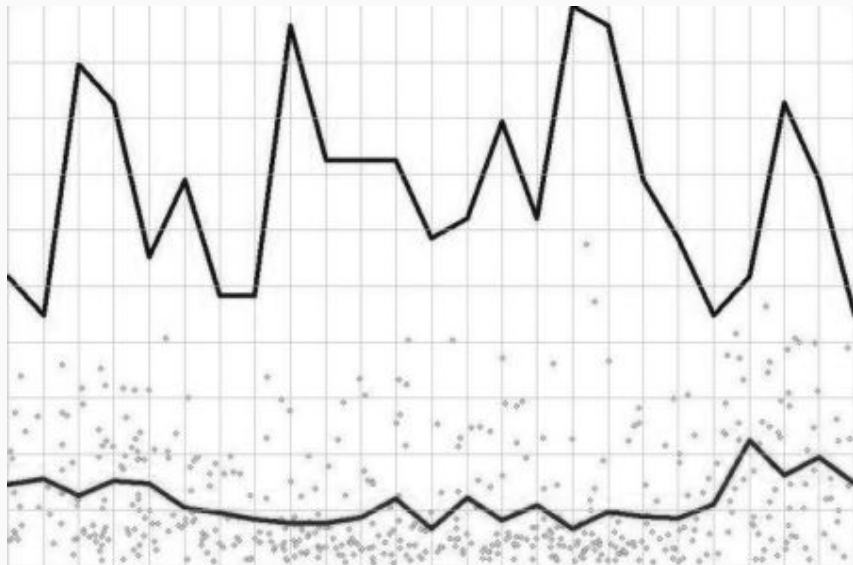
Como retirar pontos de um gráfico sem seu código?

```
library(tidyverse)

im <- image_read("C:/Users/nick_/OneDrive/Área de Trabalho/...")
im_proc <- im %>%
  image_channel("saturation")

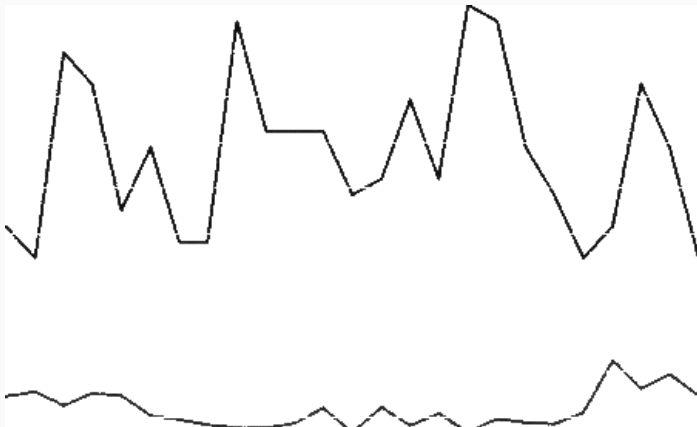
image_write(im_proc, path = "IMAGENS/grafico_ponto1.png", f
```

Como retirar pontos de um gráfico sem seu código?



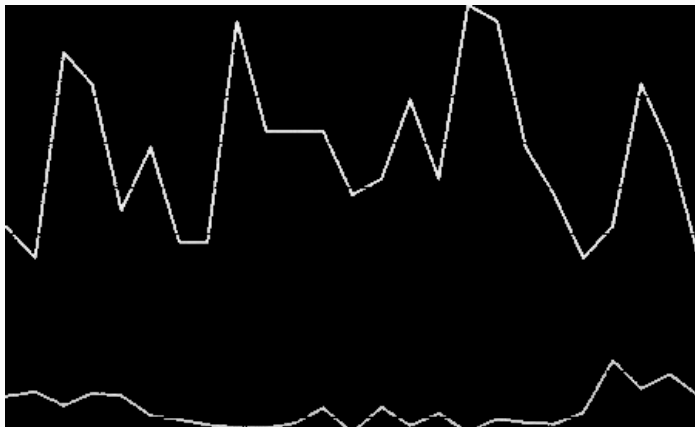
Como retirar pontos de um gráfico sem seu código?

```
im_proc2 <- im_proc %>%  
  image_threshold("white", "30%")  
image_write(im_proc2, path = "IMAGENS/grafico_ponto2.png",  
            format = "png")
```



Como retirar pontos de um gráfico sem seu código?

```
im_proc3 <- im_proc2 %>%  
  image_negate()  
image_write(im_proc3, path = "IMAGENS/grafico_ponto3.png",  
            format = "png")
```



Como retirar pontos de um gráfico sem seu código?

```
require(tidyverse)

dat <- image_data(im_proc3)[1,,] %>%
  as.data.frame() %>%
  mutate(Row = 1:nrow(.)) %>%
  select(Row, everything()) %>%
  mutate_all(as.character) %>%
  gather(key = Column, value = value, 2:ncol(.)) %>%
  mutate(Column = as.numeric(gsub("V", "", Column)),
         Row = as.numeric(Row),
         value = ifelse(value == "00", NA, 1)) %>%
  filter(!is.na(value))

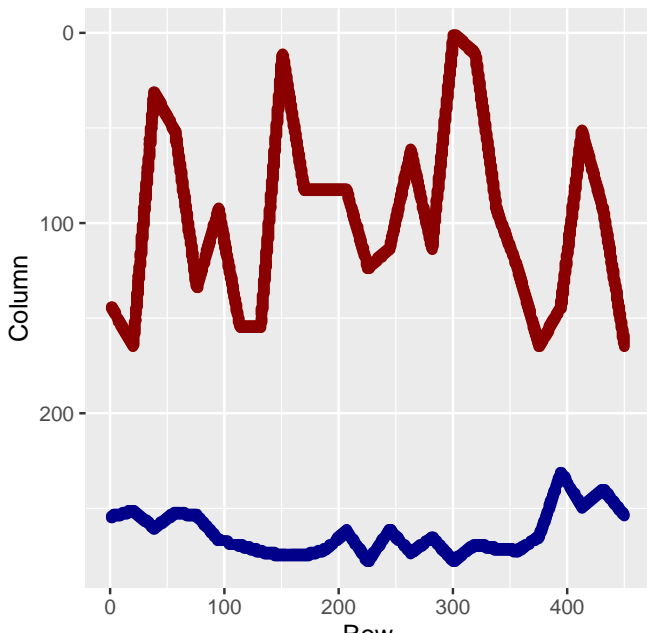
dat
```

```
##      Row Column value
```

Como retirar pontos de um gráfico sem seu código?

```
require(ggplot2)
grafico_final <-ggplot(data = dat,
                      aes(x = Row,
                          y = Column,
                          colour = (Column < 200)))) +
  geom_point() +
  scale_y_continuous(trans = "reverse") +
  scale_colour_manual(values = c( "blue4", "red4")) +
  theme(legend.position = "off")+
  ggsave("grafico_final.pdf", width = 4, height = 4)
```


Como retirar pontos de um gráfico sem seu código?



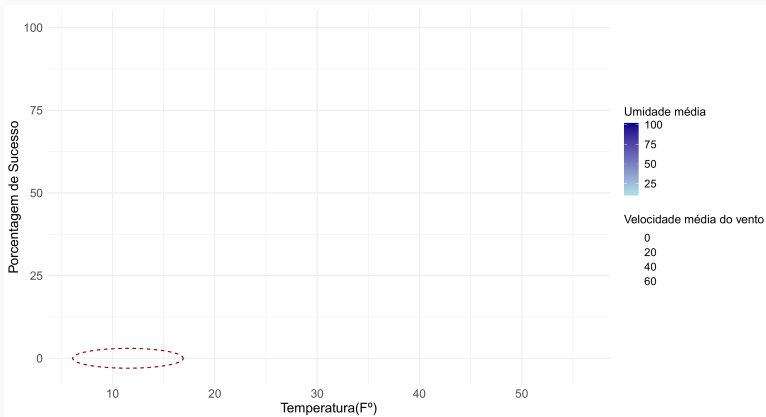
Fraqueza na leitura de PDF

```
require(pdftools)

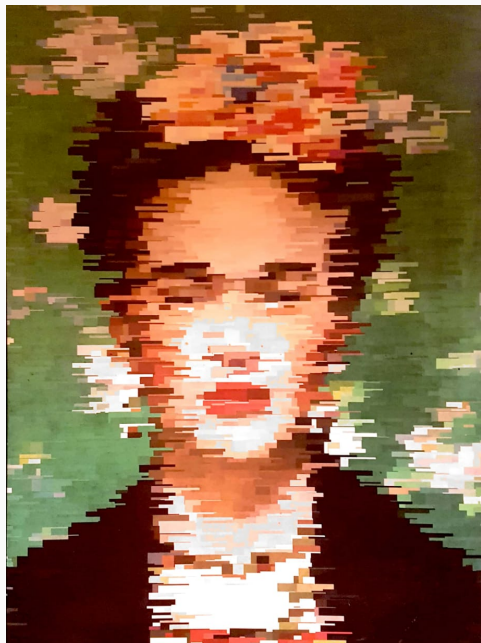
tempo <- image_read_pdf("C:/Users/nick_/OneDrive/Área de Trabalho/Tempo.pdf")

image_write(tempo, path = "tempo.pdf", format = "pdf")
```

Fraqueza na leitura de PDF



Filtro geométrico



Filtro geométrico

Por meio do pacote “imager” é possível estilizar uma imagem e deixá-la pixelizada:

```
library(imager)
foto <- load.image("C:/Users/nick_/Downloads/foto.jpg")

foto2<- foto %>% resize(size_x = 80, size_y = 80, interpolation = "nearest")
suppressMessages(suppressWarnings(library(imager)))
foto2 <- rowMeans(foto2, dims = 2)
```

Filtro geométrico

```
foto2 %>%  
  apply(1, rev) %>%  
  t() %>%  
  image(col = grey.colors(256), axes = FALSE)
```

