

Corrigindo datas

AUTHOR

Bragatte, Vital , Isaac

Carregando pacotes

```
pacman::p_load(vroom, lubridate, dplyr)
```

Criando dados fictícios

```
## Criar um vetor de datas em diferentes formatos
datas_horríveis <- c("2007-06-22", "02/27/2008", "March 14, 2009", "2010-05-31", "05/06/2011", "June 07, 2012", "2013-07-08", "08/09/2014", "September 10, 2015", "2016-10-11")

## Converter as datas para o formato de data
datas_padronizadas <- parse_date_time(datas_horríveis, orders = c("ymd", "mdy", "Bdy", "YMD", "MDY", "BYMD", "BYMDY"))

# Criar um dataframe com as datas
datas <- data.frame(
  datas_originais = datas_horríveis,
  datas_corrigidas = datas_padronizadas
)

# Visualizar o dataframe
print(datas)
```

	datas_originais	datas_corrigidas
1	2007-06-22	2007-06-22
2	02/27/2008	2008-02-27
3	March 14, 2009	2009-03-14
4	2010-05-31	2010-05-31
5	05/06/2011	2011-05-06
6	June 07, 2012	2012-06-07
7	2013-07-08	2013-07-08
8	08/09/2014	2014-08-09
9	September 10, 2015	2015-09-10
10	2016-10-11	2016-10-11

exemplo com os profes ao vivo

Chamar arquivo pessoal se quiser usar no script

```
## Colocar o nome do seu arquivo aqui - entre aspas | exemplo: hospital.csv
#suas_datas <- vroom("") #descomentar esta linha(apagar o # no começo)
```

Soluções

1. R base

Transformar em data com coluna nova no final da tabela

```
## note que a coluna datas_originais não esta como data <dtm> e sim caracter <chr>
print(datas)
```

	datas_originais	datas_corrigidas
1	2007-06-22	2007-06-22
2	02/27/2008	2008-02-27
3	March 14, 2009	2009-03-14
4	2010-05-31	2010-05-31
5	05/06/2011	2011-05-06
6	June 07, 2012	2012-06-07
7	2013-07-08	2013-07-08
8	08/09/2014	2014-08-09
9	September 10, 2015	2015-09-10
10	2016-10-11	2016-10-11

```
## Duplicar a última coluna para test
datas$novos_formato <- datas[, ncol(datas)] ##formato Datetime<dtm>
print(datas)
```

	datas_originais	datas_corrigidas	novos_formato
1	2007-06-22	2007-06-22	2007-06-22
2	02/27/2008	2008-02-27	2008-02-27
3	March 14, 2009	2009-03-14	2009-03-14
4	2010-05-31	2010-05-31	2010-05-31
5	05/06/2011	2011-05-06	2011-05-06
6	June 07, 2012	2012-06-07	2012-06-07
7	2013-07-08	2013-07-08	2013-07-08
8	08/09/2014	2014-08-09	2014-08-09
9	September 10, 2015	2015-09-10	2015-09-10
10	2016-10-11	2016-10-11	2016-10-11

```
## mudar para o padrão <date>
datas$novos_formato <- as.Date(datas$novos_formato) #date <date>
print(datas)
```

	datas_originais	datas_corrigidas	novos_formato
1	2007-06-22	2007-06-22	2007-06-22
2	02/27/2008	2008-02-27	2008-02-27
3	March 14, 2009	2009-03-14	2009-03-14
4	2010-05-31	2010-05-31	2010-05-31
5	05/06/2011	2011-05-06	2011-05-06
6	June 07, 2012	2012-06-07	2012-06-07
7	2013-07-08	2013-07-08	2013-07-08
8	08/09/2014	2014-08-09	2014-08-09
9	September 10, 2015	2015-09-10	2015-09-10
10	2016-10-11	2016-10-11	2016-10-11

```
## Mudar a ordem entre dia, mês e ano além do divisor
datas$novos_formato <- format(datas$novos_formato, format = "%d/%m/%Y")
##cada vez que mudar a ordem no R base, precisa redefinir como <date>, pois a mudança
print(datas)
```

	datas_originais	datas_corrigidas	novos_formato
1	2007-06-22	2007-06-22	22/06/2007
2	02/27/2008	2008-02-27	27/02/2008
3	March 14, 2009	2009-03-14	14/03/2009
4	2010-05-31	2010-05-31	31/05/2010
5	05/06/2011	2011-05-06	06/05/2011
6	June 07, 2012	2012-06-07	07/06/2012
7	2013-07-08	2013-07-08	08/07/2013
8	08/09/2014	2014-08-09	09/08/2014
9	September 10, 2015	2015-09-10	10/09/2015
10	2016-10-11	2016-10-11	11/10/2016

```
## round é para arredondar e deixar sem casas após a vírgula
## Sys.date pega o formato de data do R no momento, difftime como o nome já indica cal
datas$semanas <- round(as.numeric(difftime(Sys.Date(),datas$datas_corrigidas, units =
print(datas)
```

	datas_originais	datas_corrigidas	novos_formato	semanas
1	2007-06-22	2007-06-22	22/06/2007	16
2	02/27/2008	2008-02-27	27/02/2008	15
3	March 14, 2009	2009-03-14	14/03/2009	14
4	2010-05-31	2010-05-31	31/05/2010	13
5	05/06/2011	2011-05-06	06/05/2011	12
6	June 07, 2012	2012-06-07	07/06/2012	11
7	2013-07-08	2013-07-08	08/07/2013	10
8	08/09/2014	2014-08-09	09/08/2014	9
9	September 10, 2015	2015-09-10	10/09/2015	8
10	2016-10-11	2016-10-11	11/10/2016	7

2. Lubridate

Necessário somente uma linha para editar campos que já são datas

```
## Converter a coluna 'novos_formato' para uma data <date>
datas$novos_formato <- dmy(datas$novos_formato) #linha1
print(datas)
```

	datas_originais	datas_corrigidas	novos_formato	semanas
1	2007-06-22	2007-06-22	2007-06-22	16
2	02/27/2008	2008-02-27	2008-02-27	15
3	March 14, 2009	2009-03-14	2009-03-14	14
4	2010-05-31	2010-05-31	2010-05-31	13
5	05/06/2011	2011-05-06	2011-05-06	12

6	June 07, 2012	2012-06-07	2012-06-07	11
7	2013-07-08	2013-07-08	2013-07-08	10
8	08/09/2014	2014-08-09	2014-08-09	9
9	September 10, 2015	2015-09-10	2015-09-10	8
10	2016-10-11	2016-10-11	2016-10-11	7

Extras com lubridate

```
## Extrair o ano, mês e dia da coluna 'nascimento' em novas colunas
datas <- datas |>
  mutate(
    ano_nascimento = year(novo_formato),
    mes_nascimento = month(novo_formato),
    dia_nascimento = day(novo_formato)
  )
print(datas)
```

	datas_originais	datas_corrigidas	novo_formato	semanas	ano_nascimento
1	2007-06-22	2007-06-22	2007-06-22	16	2007
2	02/27/2008	2008-02-27	2008-02-27	15	2008
3	March 14, 2009	2009-03-14	2009-03-14	14	2009
4	2010-05-31	2010-05-31	2010-05-31	13	2010
5	05/06/2011	2011-05-06	2011-05-06	12	2011
6	June 07, 2012	2012-06-07	2012-06-07	11	2012
7	2013-07-08	2013-07-08	2013-07-08	10	2013
8	08/09/2014	2014-08-09	2014-08-09	9	2014
9	September 10, 2015	2015-09-10	2015-09-10	8	2015
10	2016-10-11	2016-10-11	2016-10-11	7	2016
	mes_nascimento	dia_nascimento			
1	6	22			
2	2	27			
3	3	14			
4	5	31			
5	5	6			
6	6	7			
7	7	8			
8	8	9			
9	9	10			
10	10	11			

Calculando idade com a data de hoje

```
idade_presidente <- (today() - ymd(19451027))/365.5

idade_presidente
```

Time difference of 77.658 days