INF-612 Handling Data

Paulo Sigrist / Rafael Sangalli

October 7, 2015

Carregando os dados:

O primeiro passo é carregar os dados no formato CSV. Nesse processo, o período já é filtrado para o período desejado:

```
cpa = loadAndFilterData()
```

Abaixo, o resultado com os primeiros registros:

	horario	temperatura	vento	umidade	sensacao
16313	2014-07-01 00:00:00	12.7	15.2	73.6	8.0
16314	2014-07-01 00:10:00	12.6	13.7	73.6	8.2
16315	2014-07-01 00:20:00	12.5	12.9	74.0	8.2
16316	2014-07-01 00:30:00	12.6	11.3	73.7	8.6
16317	2014-07-01 00:40:00	12.8	6.3	72.7	9.7
16318	2014-07-01 00:50:00	13.4	7.0	70.5	10.2

E agora os últimos registros:

	horario	temperatura	vento	umidade	sensacao
68013	2015-06-30 23:00:00	17.7	20.7	81.7	16.2
68014	2015-06-30 23:10:00	17.5	17.8	82.4	16.0
68015	2015-06-30 23:20:00	17.4	16.2	82.6	15.9
68016	2015-06-30 23:30:00	17.1	16.7	83.6	15.5
68017	2015-06-30 23:40:00	17.2	17.3	83.0	15.7
68018	2015-06-30 23:50:00	17.2	17.9	83.4	15.6

Análises:

Após a obteção dos dados, fizemos as seguintes análises:

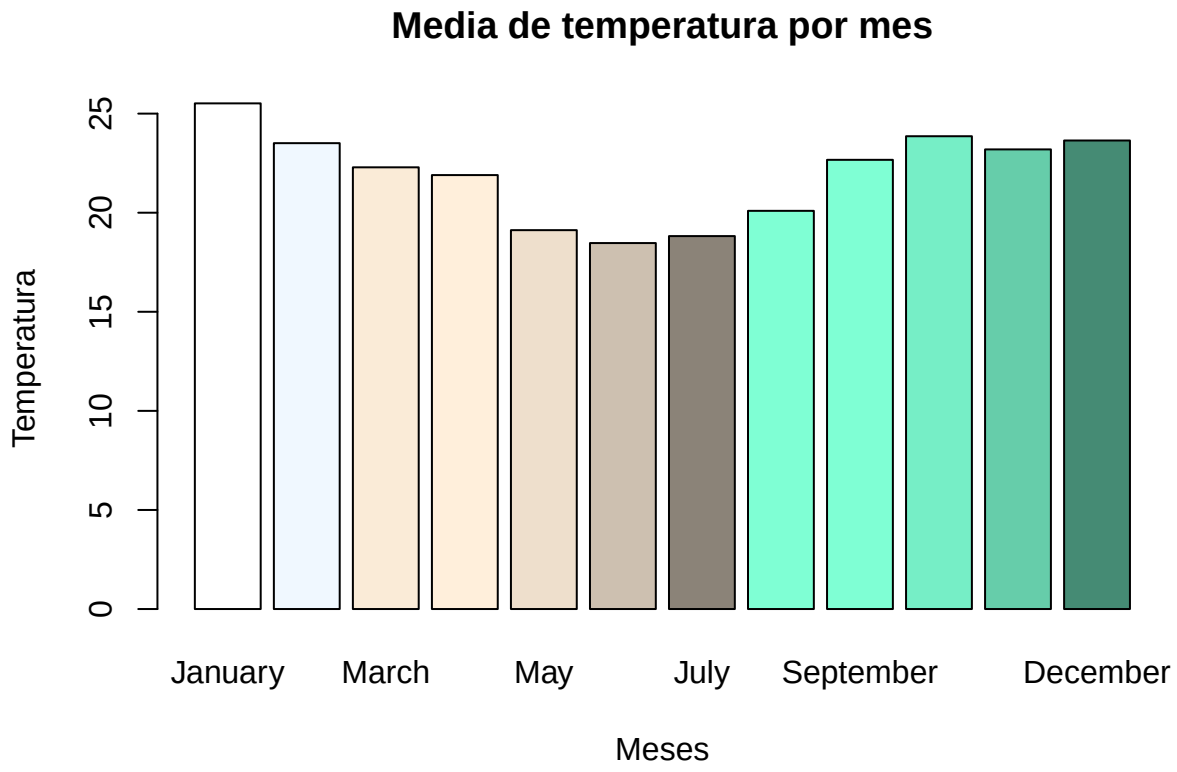
- Média de temperatura por mês
- Comparadação da evolução da temperatura por dia a cada mês
- Histograma das medidas por estação do ano

Média de temperatura por mes:

Nessa análise, utilizamos a função `tapply` para calcular a média da temperatura por mês. Utilizamos o parametro `na.rm = TRUE` para ignorar valores *NA*. Para adicionar o nome dos meses no resultado do cálculo, utilizamos a função `months` e atribuimos ao `rownames` do resultado. Abaixo está o código completo da função `meanByMonth`:

```
## function (cpa)
## {
##     r <- tapply(cpa[["temperatura"]], cpa[["horario"]]$mon, mean,
##         na.rm = TRUE)
##     rownames(r) <- month.name
##     barplot(r, col = colors()[1:12], main = "Media de temperatura por mes",
##         xlab = "Meses", ylab = "Temperatura")
## }
```

Podemos observar nesse gráfico que a temperatura cai por volta de 5 graus nos meses de inverno.



Evolução da temperatura dia após dia:

Nessa análise, podemos observar a evolução da temperatura por dia, a cada mês.

Para efetuar esse cálculo, foram criadas duas funções auxiliares, `dataByMonth` e `meanByDay`. A primeira irá retornar somente dados do `cpa` do mês especificado. A segunda irá calcular a média da temperatura por dia. Abaixo o código das duas funções:

```
dataByMonth
```

```
## function (m, cpa)
## {
##     cpa[cpa$horario$mon == m, ]
## }
```

```
meanByDay
```

```
## function (month, cpa)
## {
##     x <- dataByMonth(month, cpa)
##     tapply(x[["temperatura"]], x[["horario"]]$mday, mean, na.rm = TRUE)
## }
```

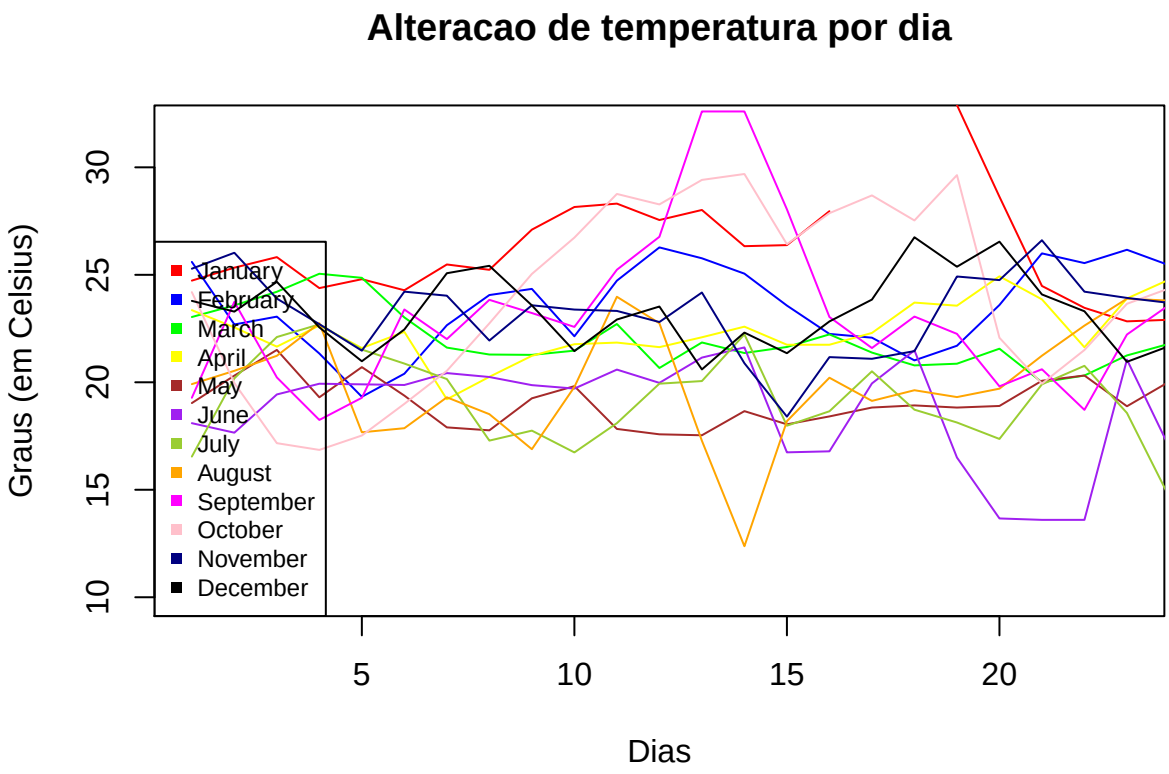
Agora podendo calcular a média de temperatura por dia para cada mês, temos que aplicar isso no nosso data frame `cpa`. Para isso, criamos um array de 0 até 11 e com a função `sapply` calculamos o resultado.

Plotamos um gráfico e depois novamente utilizando a função `sapply` adicionamos as linhas dos resultados no gráfico. Abaixo o código fonte da função:

```
temperatureByDay
```

```
## function (cpa)
## {
##     r <- sapply(c(0:11), meanByDay, cpa)
##     cores <- c("red", "blue", "green", "yellow", "brown", "purple",
##         "yellowgreen", "orange", "magenta", "pink", "navy", "black")
##     plot(c(10:32), type = "n", main = "Alteracao de temperatura por dia",
##         xlab = "Dias", ylab = "Graus (em Celsius)")
##     sapply(c(1:12), function(l) {
##         lines(unlist(r[l]), col = cores[l])
##     })
##     legend("bottomleft", month.name, col = cores, pch = 15, cex = 0.75)
## }
```

O resultado final é o gráfico abaixo:



Histogramas por estação do ano:

Nessa análise, plotamos um historiograma para cada medida obtida (temperatura, sensação, vento e umidade) para cada estação do ano. Para agrupar os dados por estação, foi criada uma função chamada `dataBySeason` onde criamos uma estrutura que contém os dados agrupados pelas quatro estações do ano. Abaixo está o código da função:

```
dataBySeason
```

```
## function (cpa)
## {
##     inverno <- dataByInterval("2014-06-21 00:00:00", "2014-09-21 23:59:59",
##         cpa)
##     primavera <- dataByInterval("2014-09-22 00:00:00", "2014-12-20 23:59:59",
##         cpa)
##     verao <- dataByInterval("2014-12-22 00:00:00", "2015-03-19 23:59:59",
##         cpa)
##     outono <- dataByInterval("2015-03-20 00:00:00", "2015-06-20 23:59:59",
##         cpa)
##     inverno2 <- dataByInterval("2015-06-21 00:00:00", "2015-09-21 23:59:59",
##         cpa)
##     inverno <- rbind(inverno, inverno2)
##     m <- 0
##     m$inverno <- inverno
##     m$primavera <- primavera
```

```
##     m$verao <- verao
##     m$outono <- outono
##     return(m)
## }
```

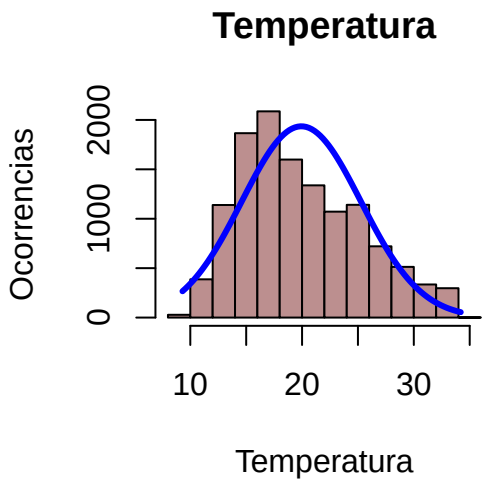
Foi criada uma função genérica para plotar um historiograma de um array de dados, onde passamos alguns valores, como o array de dados, título do gráfico, label do eixo X e cores do gráfico. Abaixo está o código da função `histogramByValue`:

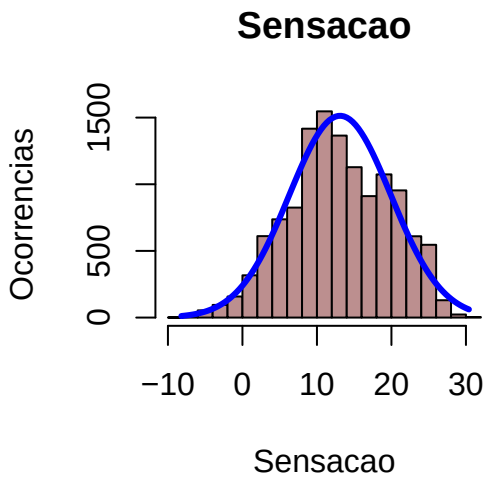
```
histogramByValue
```

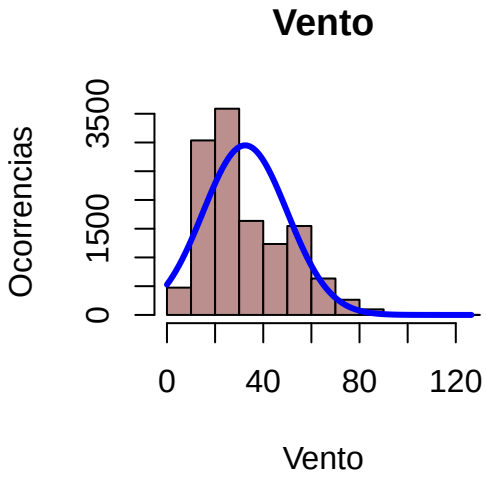
```
## function (data, title, xlabel = title, col1 = "rosybrown", col2 = "blue")
## {
##     h <- hist(data, main = title, xlab = xlabel, ylab = "Ocorrencias",
##         col = col1)
##     xfit <- seq(min(data, na.rm = TRUE), max(data, na.rm = TRUE),
##         length = 100)
##     yfit <- dnorm(xfit, mean = mean(data, na.rm = TRUE), sd = sd(data,
##         na.rm = TRUE))
##     yfit <- yfit * diff(h$mids[1:2]) * length(data)
##     lines(xfit, yfit, col = col2, lwd = 3)
## }
```

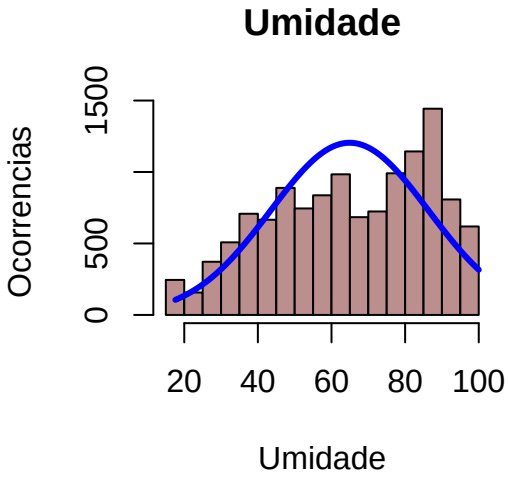
Com essas duas funções, podemos agora os histogramas para análise

Inverno



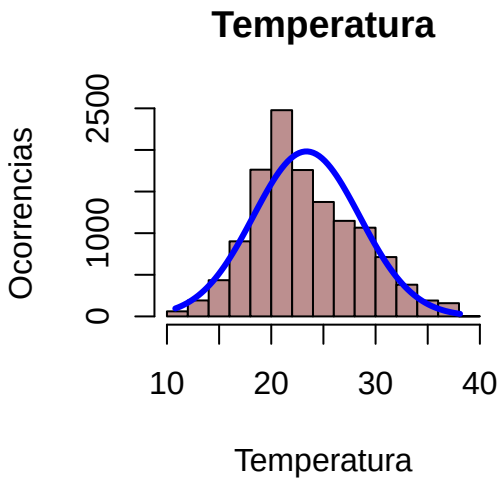


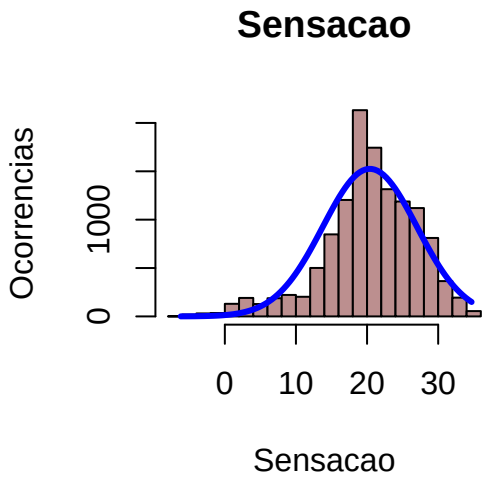


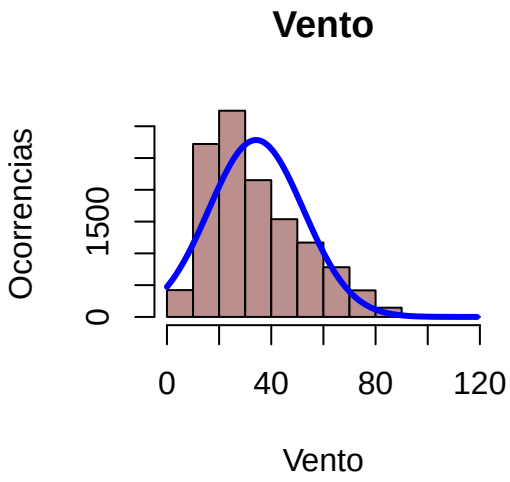


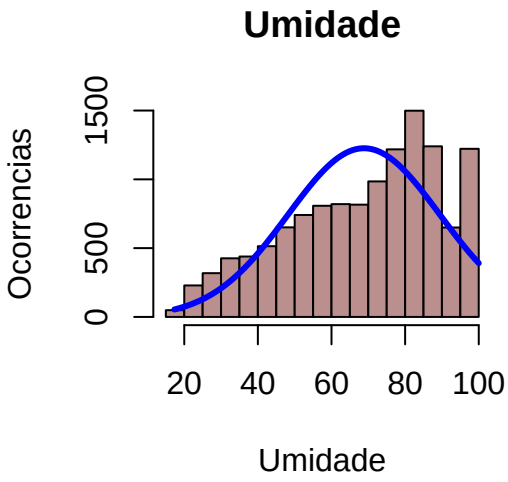
Podemos observar que no inverno, a temperatura se concentrou entre 15 e 20 graus e a sensação ficou um pouco abaixo, entre 10 e 15 graus e tivemos uma umidade alta.

Primavera



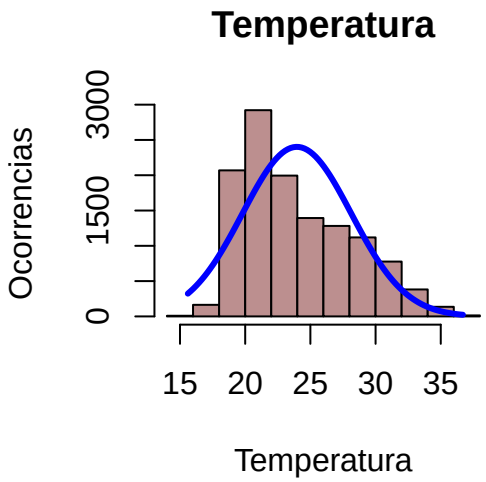


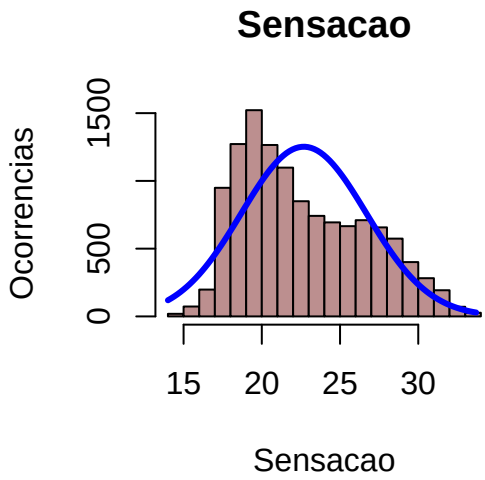


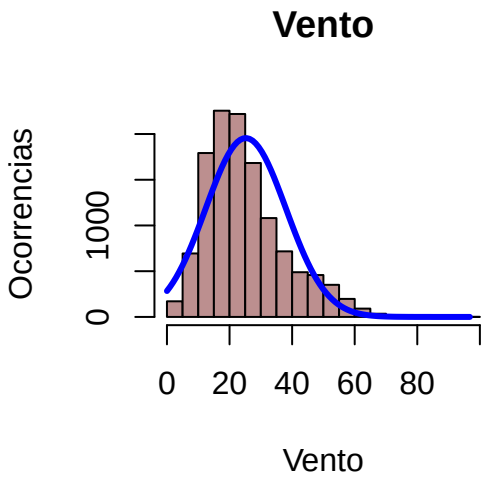


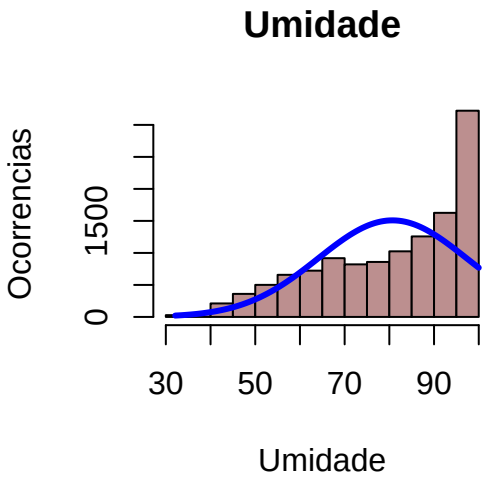
Já na primavera, a temperatura foi mais elevada, ficando entre 20 e 25 graus por mais vezes. Já a sensação ficou entre 18 e 23 graus.

Verão



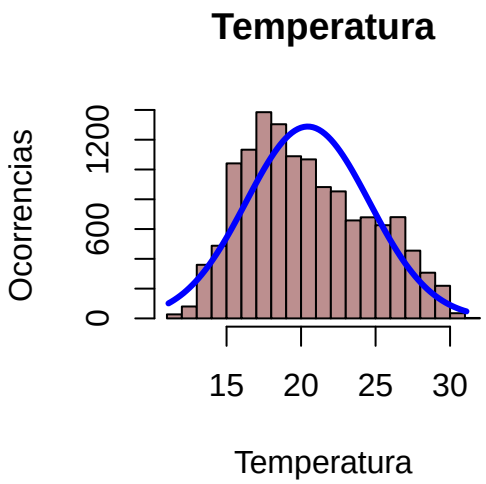


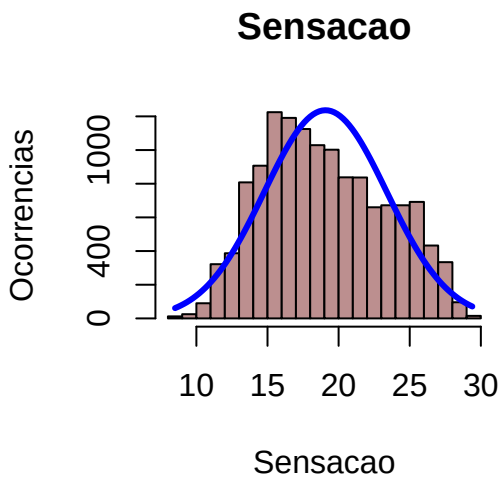


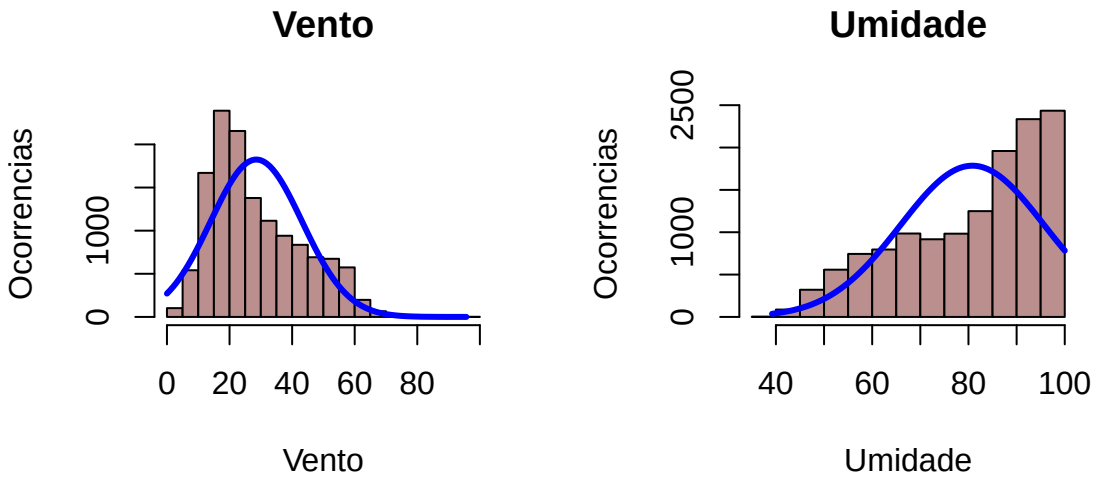


A temperatura do verão foi um pouco mais alta que a primavera, e ficou entre 25 e 30 graus, mas com uma umidade bem alta.

Outono







No outono já tivemos temperaturas mais baixas, concentrando sua variação entre 15 e 22 graus e uma umidade bem alta.

Conclusão

Para poder fazer análises dos dados em *R*, a maior parte do trabalho está em preparar os dados para serem analisados. Os maiores desafios encontrados foram conversão de tipos de dados (*string* para inteiro, *string* para data), agrupar os dados de forma que faça sentido (exemplo, agrupar os dados por estações do ano), tratar dados que estão faltando (valores *NA*)

Depois que o prepara dos dados foi finalizada, a criação dos gráficos foi mais simples, tendo apenas que tomar cuidado com a sintaxe que cada tipo de gráfico precisa.