Análise exploratória de Dados

Fonte desse material: https://rawgit.com/mhahsler/Introduction_to_Data_Mining_R_Examples/master/chap2_exploring.html E https://rawgit.com/mhahsler/Introduction_to_Data_Mining_R_Examples/master/chap2.html E ... (alguns outros)

** FAÇA PASSO A PASSO CADA ETAPA DESTE DOCUMENTO, ENTENDA O COMANDO E A RESPECTIVA SAIDA, BUSQUE A DOCUMENTAÇÃO DAS BIBLIOTECAS USADAS EM CASO DE DUVIDA SOBRE OS PARAMETROS E FORMATOS DE CADA COMANDO.

** NÃO FAÇA SIMPLESMENTE UM COPY+PASTE. PARA A ATIVIDADE DA SEMANA VOCÊ VAI PRECISAR TER ENTENDIDO BEM TUDO QUE TEM NESSE DOCUMENTO.

Talvez voce precise antes instalar as bibliotecas necessarias para realizar essa atividade.

Instale as bibliotecas: tidyverse e ggplot2.

Caso voce não tenha instalado ainda, descomente as linhas do codigo R abaixo.

```
#install.packages("tidyverse")
#install.packages("ggplot2")
#install.packages("GGally")
#install.packages("ggcorrplot")
```

A seguir, precisamos chamar essas bibliotecas, para que suas funções sejam utilizadas:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggcorrplot)
```

Estatística básica sobre os dados

A exploração dos dados é importante para conhecer os atributos que você vai trabalhar. Que tipo eles são? como se distribuem? estão corretos e completos? Antes de qualquer tarefa de mineração de dados, é importante que se tenha conhecimento sobre o conjunto de dados que vai trabalhar. Esse conhecimento pode

ser junto com o especialista, sobre o significado de cada variável e/ou um conhecimento mais técnico sobre os valores de determinado atributo e seus caracteristicas.

Como o exemplo, vamos usar a base de dados Iris. A base de dados IRIS é um conjunto de dados sobre flores do tipo IRIS que podem ser de 3 espécies: iris setosa, iris versicolor e iris virginica. Sobre cada flor, foram obtidos dados sobre comprimento e largura de pétalas e sépalas. De acordo com esses atributos, um especialista classificou as flores nas 3 possíveis classes. A partir desse conjunto de dados rotulado, é então possível treinar e obter um modelo preditivo (aprendizado supervisionado, pois temos um atributo-alvo, a espécie). A base de dados IRIS é muito utilizada em atividades didáticas para quem está começando a trabalhar com Ciência de Dados.

Para a atividade proposta da semana voce vai gerar um documento similar a esse, porém com a base de dados indicada e seguindo as instruções da atividade proposta. Esse documento aqui é um estudo dirigido, não precisa ser enviado no AVA.

```
# a base de dados iris já faz parte do conteudo do R, entao é só chama-la
# para a tarefa voce precisará abrir o arquivo CSV
data(iris)
#convert the data.frame into a tidyverse tibble (optional)
#é um tipo de dataframe simples
iris <- as_tibble(iris)
#se chama o nome da base de dados para mostrar as primeiras instancias de dados
iris
```

```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            1.4         0.2 setosa
##  3          4.7         3.2          1.3         0.2 setosa
##  4          4.6         3.1          1.5         0.2 setosa
##  5          5           3.6          1.4         0.2 setosa
##  6          5.4         3.9          1.7         0.4 setosa
##  7          4.6         3.4          1.4         0.3 setosa
##  8          5           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```

Olhando a saida desse comando (iris, que corresponde ao nome da variavel do tipo dataframe que recebeu esses dados) voce já começa a conhecer esses dados: são 5 atributos: Sepal.Length Sepal.Width Petal.Length Petal.Width Species Todos os atributos são do tipo (double). Também voce pode ver que a tabela tem 150 linhas, que correspondem a 150 instâncias.

Para obter estatisticas básicas sobre os atributos, use o comando "summary":

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
```

```
##  virginica :50
##
##
##
```

Analisando o resultado desse comando: Vemos os valores mínimo, mediana, média, máximo primeiro e terceiro quartil para cada variavel númerica (os 4 atributos preditivos) e para o atributo categórico, Species, temos o total de exemplos de cada classe.

**Quartis (Q1, Q2 e Q3): São valores dados a partir do conjunto de observações ordenado em ordem crescente, que dividem a distribuição em quatro partes iguais. O primeiro quartil, Q1, é o número que deixa 25% das observações abaixo e 75% acima, enquanto que o terceiro quartil, Q3, deixa 75% das observações abaixo e 25% acima. Já Q2 é a mediana, deixa 50% das observações abaixo e 50% das observações acima.

Outra medida importante a ser utilizada na avaliação de atributos numéricos é o desvio padrão. Obtendo média e desvio padrão para o atributo: sepal length

```
iris %>% pull(Sepal.Length) %>% mean()
```

```
## [1] 5.843333
```

```
iris %>% pull(Sepal.Length) %>% sd()
```

```
## [1] 0.8280661
```

Calculando um sumário de média (mean) , desvio padrão (sd)

```
iris %>% summarize_if(is.numeric, mean)
```

```
## # A tibble: 1 x 4
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
##          <dbl>       <dbl>        <dbl>       <dbl>
## 1         5.84        3.06         3.76        1.20
```

```
iris %>% summarize_if(is.numeric, sd)
```

```
## # A tibble: 1 x 4
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
##          <dbl>       <dbl>        <dbl>       <dbl>
## 1        0.828       0.436         1.77       0.762
```

Para atributos categóricos como Species, é interessante saber o total de instancias de cada classe. Para isso, além do summary, voce pode usar o count.

```
iris %>% count(Species)
```

```
## # A tibble: 3 x 2
##   Species        n
##   <fct>      <int>
## 1 setosa        50
## 2 versicolor    50
## 3 virginica     50
```

** No material original tem outros comandos que vamos discutir na semana que vem, já correspondem a pré-processamento.

Manipulação dos dados

Para manipulação dos dados temos várias funções. Uma delas é o comando select(). Pode ser usado para, por exemplo, criar subconjuntos dos dados. Ela seleciona dados pelos nomes das colunas e voce pode selecionar diferentes numeros de colunas de diferentes formas.

```
#selecionando somente dados sobre as sepalas mais o atributo preditivo.
somentesepala<-select(iris,Sepal.Length, Sepal.Width,Species)
summary(somentesepala)
```

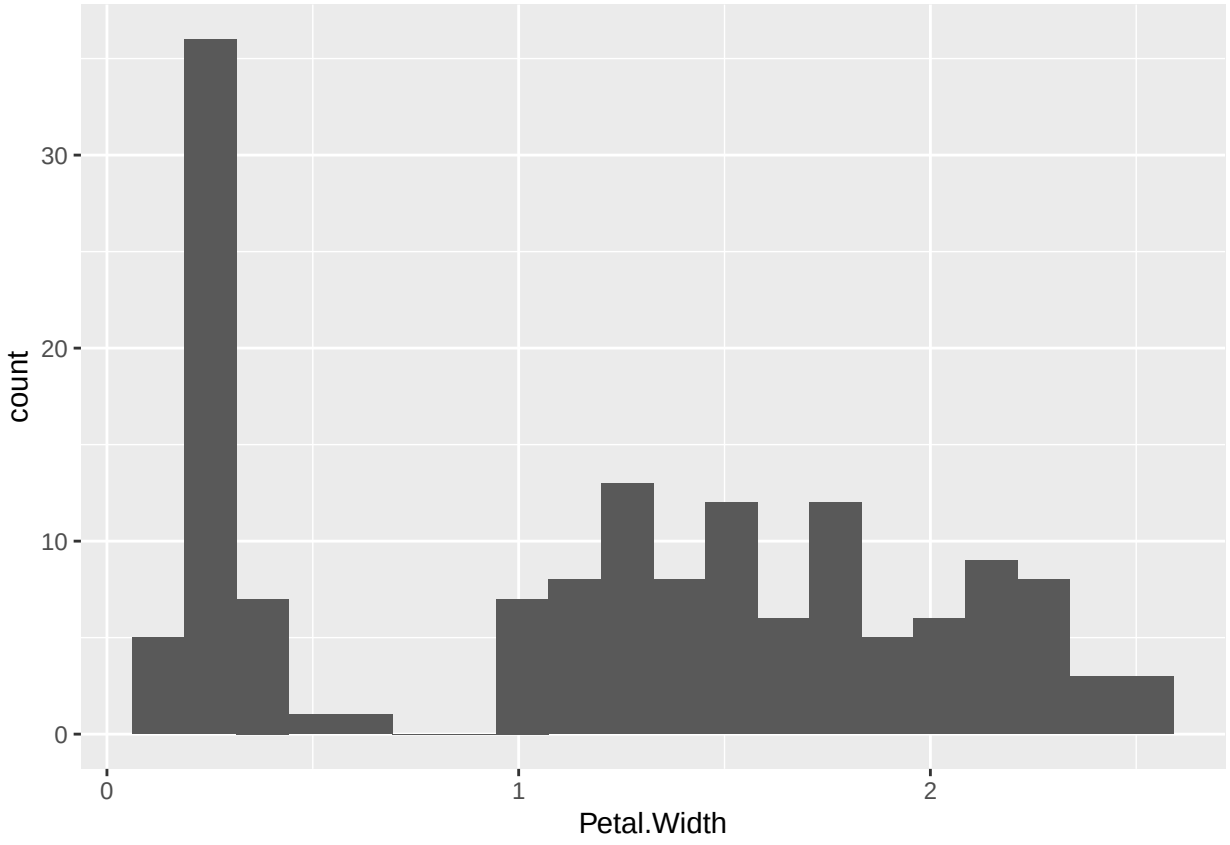
```
##   Sepal.Length    Sepal.Width          Species
##  Min.   :4.300   Min.   :2.000   setosa    :50
##  1st Qu.:5.100   1st Qu.:2.800   versicolor:50
##  Median :5.800   Median :3.000   virginica :50
##  Mean   :5.843   Mean   :3.057
##  3rd Qu.:6.400   3rd Qu.:3.300
##  Max.   :7.900   Max.   :4.400
```

Visualizações - Plotando graficos com ggplot

Histograma:

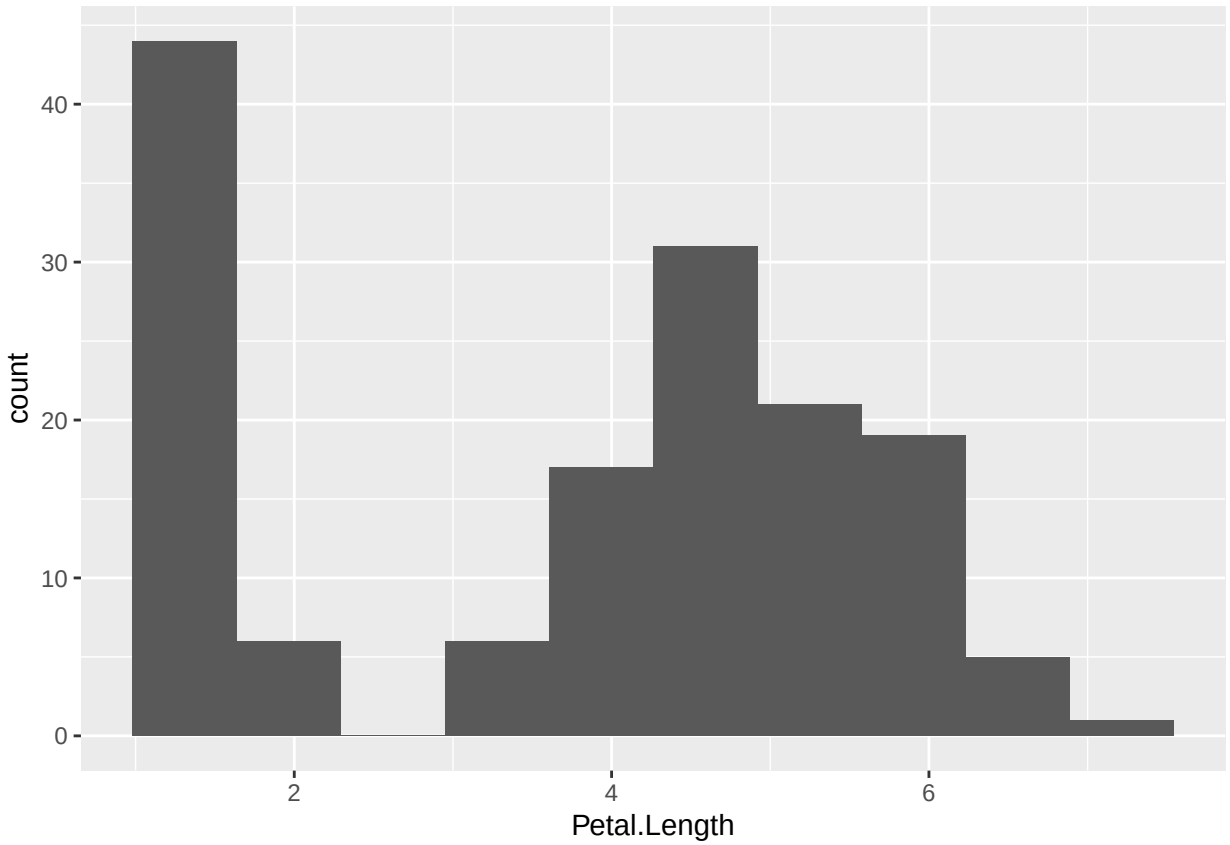
Histograma da variável Petal.width. escolhemos particionar os valores desse atributos em 20 faixas de valores que serão igualmente divididas entre o valor mínimo e máximo. Note que os valores para esse atributo vão de 0.1 até 2.5 (veja o resultado de summary).

```
ggplot(iris, aes(Petal.Width)) + geom_histogram(bins = 20)
```



Um segundo exemplo é o histograma da variável Petal.Length. onde decidimos particionar os valores desse atributos em 10 faixas de valores que serão igualmente divididas entre o valor mínimo e máximo.

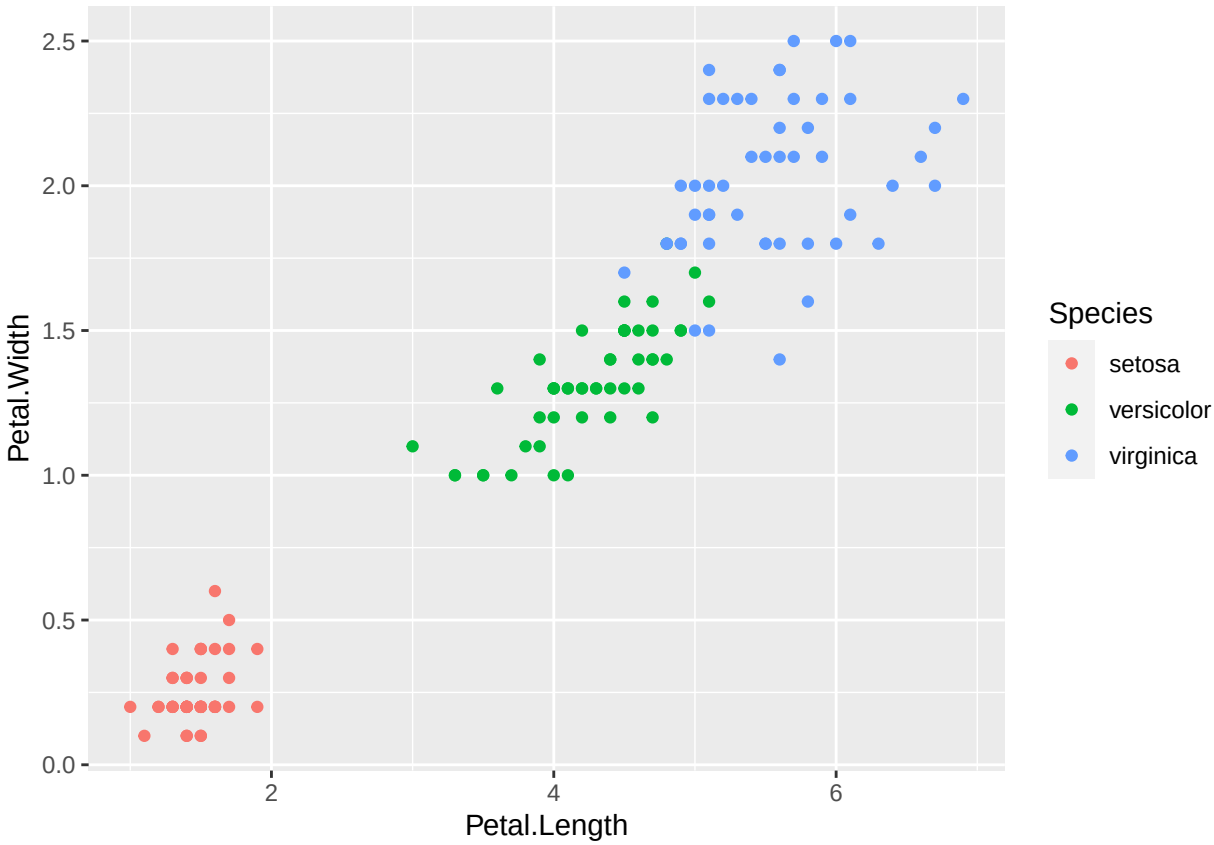
```
ggplot(iris, aes(Petal.Length)) + geom_histogram(bins = 10)
```



Scatter plot

Esse grafico permite gerar análises de relação entre 2 variaveis numericas, e podemos usar o recurso da cor, para incluir o nosso atributo alvo, permitindo visaulizar a relação entre 3 variaveis em um grafico.

```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) + geom_point()
```



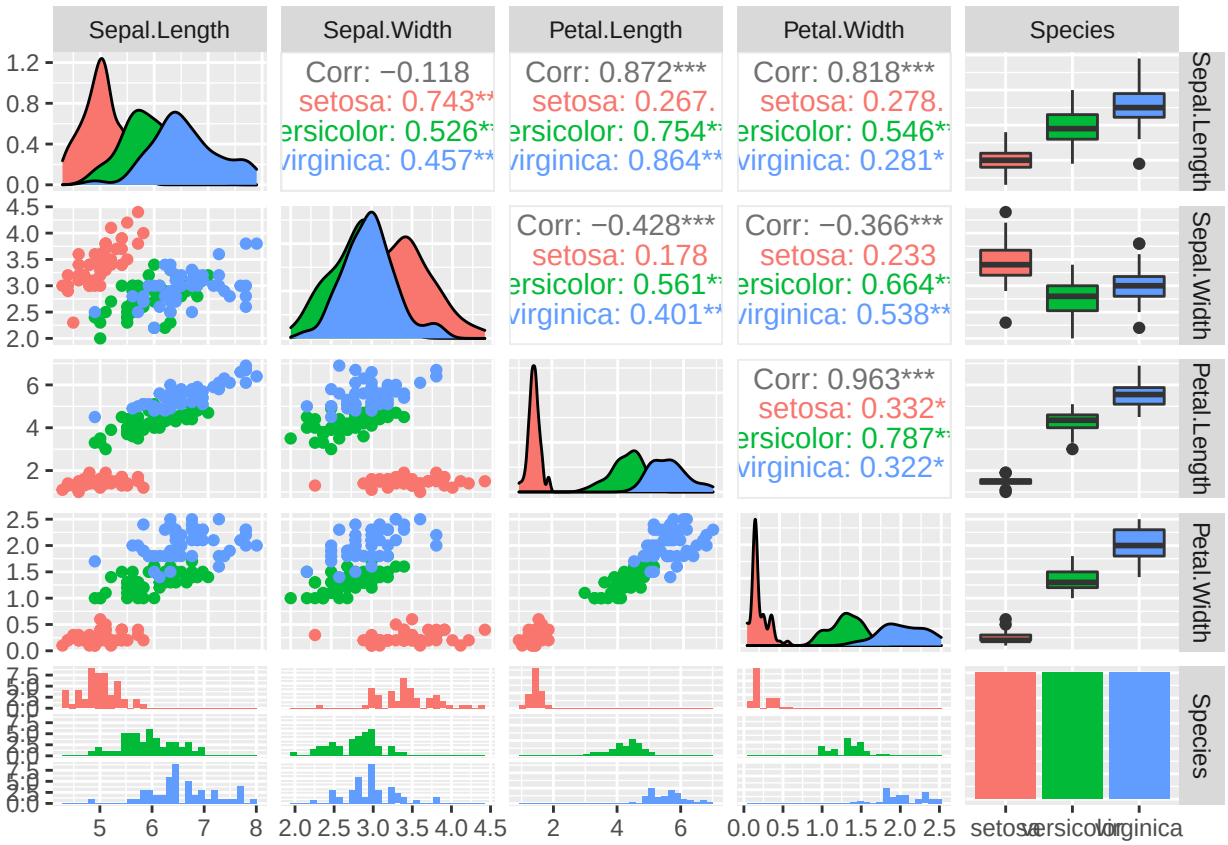
Analisando esse gráfico por meio de uma inspeção visual, podemos ver que todas as instâncias cuja os valores de largura e tamanho da petala são pequenos corresponde a classe de flores Iris Setosa. O mesmo se observa com os valores médios, todos são Iris Versicolor, e os maiores valores para ambos os atributos correspondem a classe Iris Viriginica.

Scatter plot matrix

Expandido essa análise de dados, vamos fazer uma grande matrix, que inclui vários graficos para analisar variaveis numéricas. Você pode usar esse gráfico para analisar a qualidade dos dados, pois tem uma visão geral. Você pode escolher plotar apenas alguns dos atributos que seja de seu interesse inspecionar com mais atenção.

```
library("GGally")
ggpairs(iris,  aes(color=Species))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



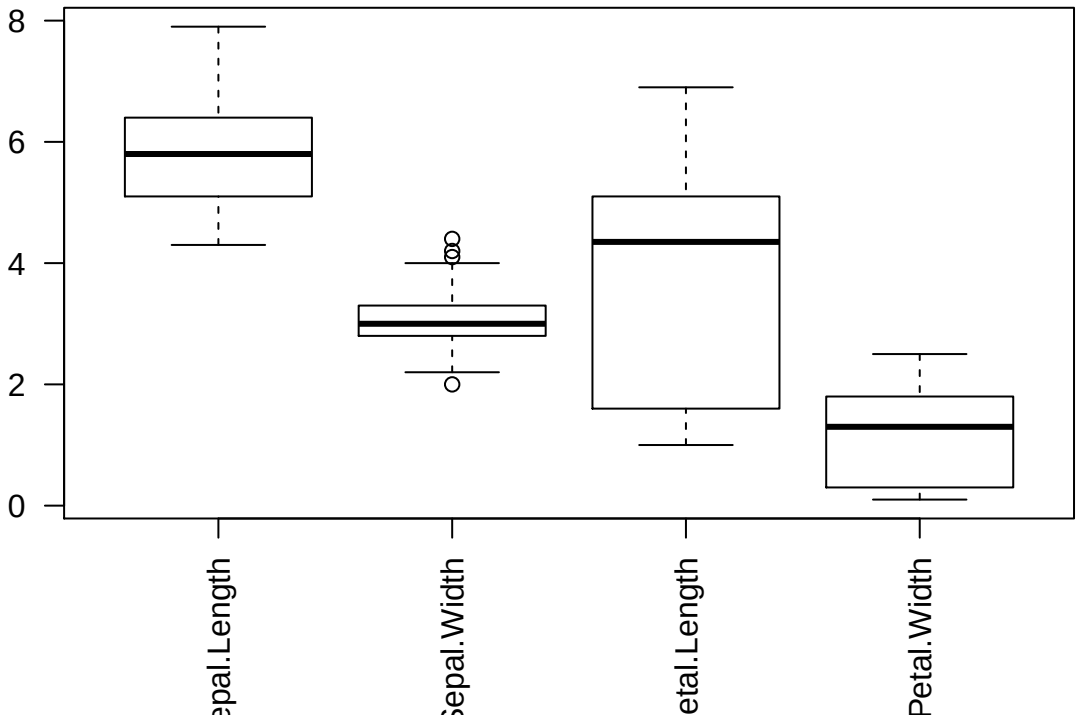
Neste gráfico, as cores estão sempre relacionadas com as classes das flores.

BoxPlot

Permite a comparação de distribuição de variaveis continuas. Se quiser se aprofundar sobre a interpretação de um bloxplot há muito material disponivel. Uma sugestão: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

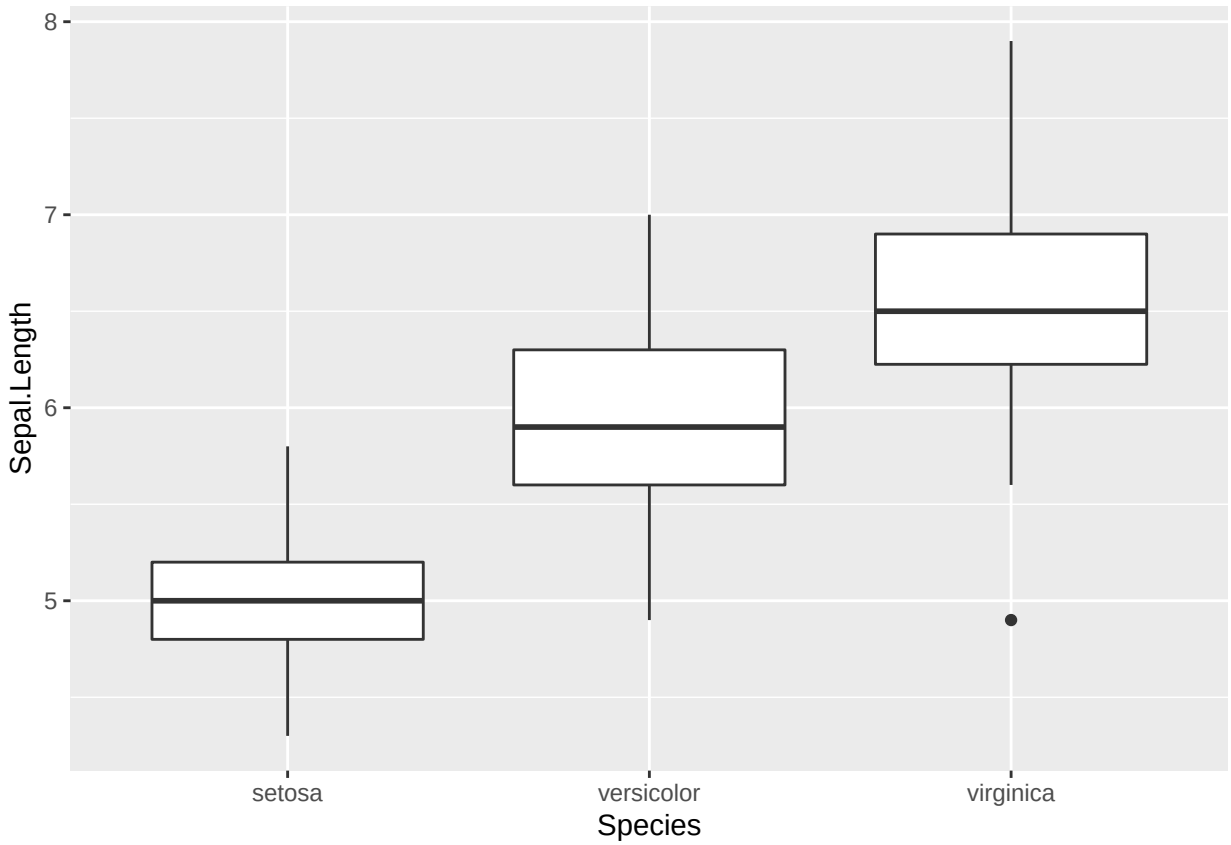
Podemos fazer um bloxplot bem geral, olhando todos os atributos preditivos:

```
boxplot(iris[,1:4],las=2)
```



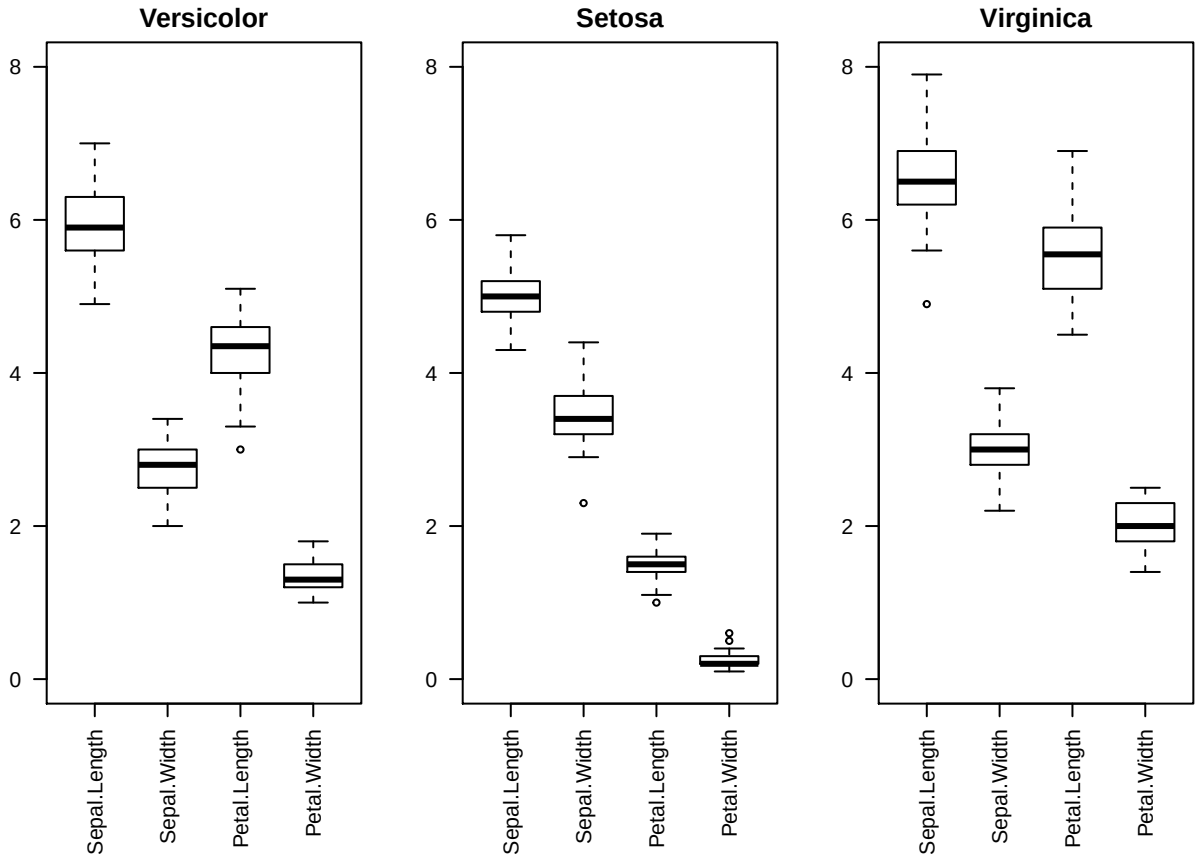
Ou bloxplos especificos de determinado atributo preditivo, separado de acordo com o valor do atributo alvo. Nesse caso podemos ver uma diferença clara de distribuição dos valores do atributo Sepal.Length de acordo com a especie (e também uma variação maior de valores para as classes versicolor e virginica).

```
ggplot(iris, aes(Species, Sepal.Length)) + geom_boxplot()
```



Expandindo essa ideia de ver por classe, podemos criar subconjuntos (como o comando subset), criando bases especificas por classe. A seguir, são exibidos 3 bloxplots, de acordo com a classe, das 3 variaveis. Desse forma podemos analisar de forma geral as diferenças entre as classes de acordo com os valores dos 4 atributo preditivos.

```
irisVer <- subset(iris, Species == "versicolor")
irisSet <- subset(iris, Species == "setosa")
irisVir <- subset(iris, Species == "virginica")
par(mfrow=c(1,3),mar=c(6,3,2,1))
boxplot(irisVer[,1:4], main="Versicolor",ylim = c(0,8),las=2)
boxplot(irisSet[,1:4], main="Setosa",ylim = c(0,8),las=2)
boxplot(irisVir[,1:4], main="Virginica",ylim = c(0,8),las=2)
```

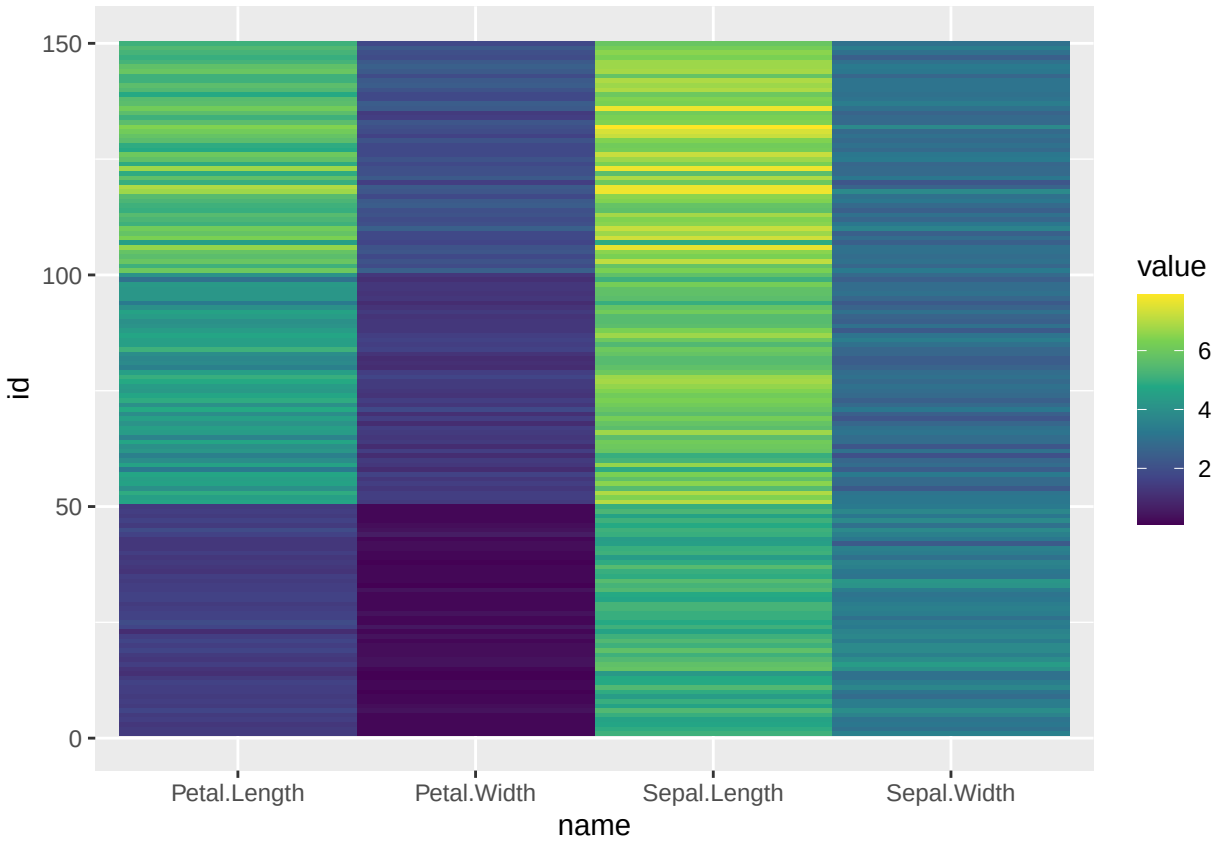


Data matrix visualization

É possível utilizar comando do ggplot para visualizar os dados no formato de matrizes, coloridos por alqum(ou alguns) dos valores dos atributos. Esse tipo de visualização é mais interessante de visualizar um ou 2 atributos relacionados e com faixas de valores que sejam relacionadas.

Visualizando os 4 atributos numéricos.

```
ggplot(iris %>% mutate(id = row_number()) %>% pivot_longer(cols = 1:4),
  aes(x = name, y = id, fill = value)) + geom_tile() +
  scale_fill_viridis_c()
```

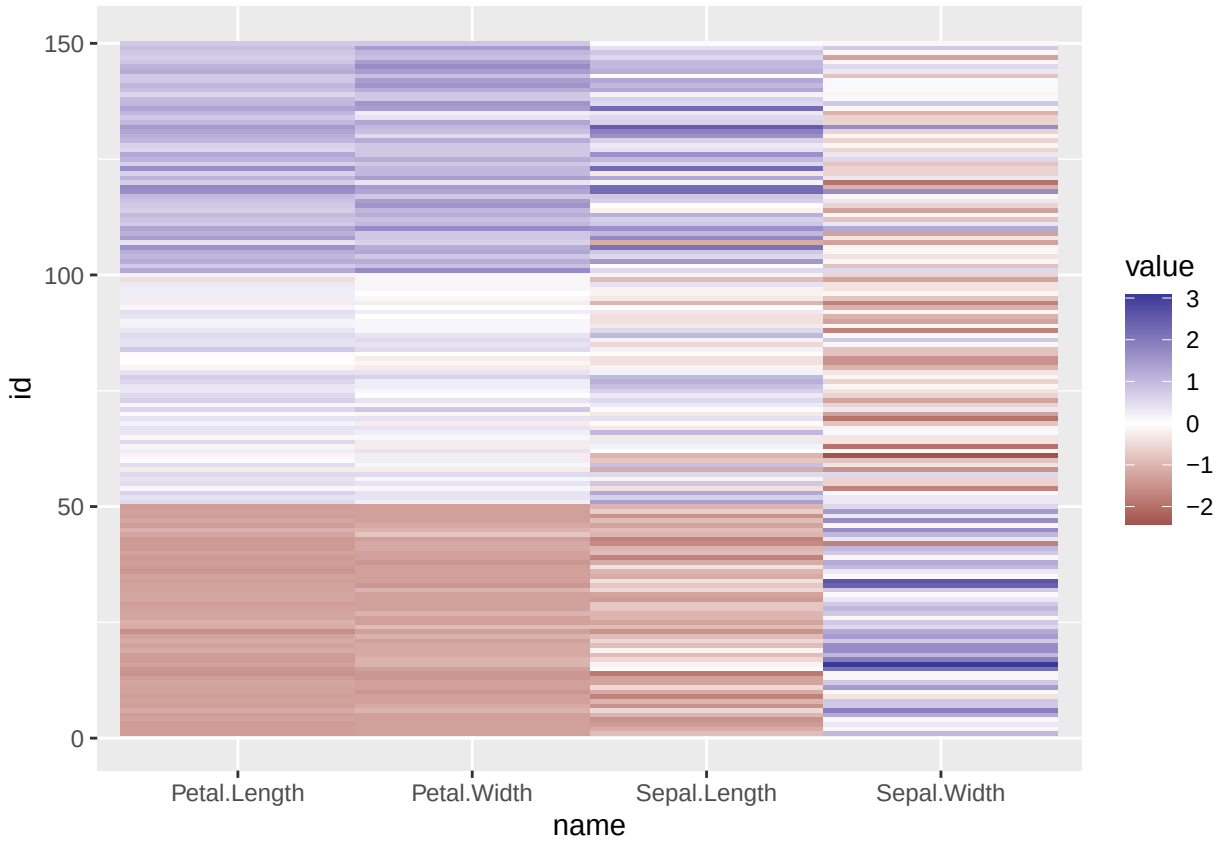


Analisando o atributo Petal.width, conforme já tinhamos discutido nesse documento, seus valores variam de 0.1 até 2.5. O que justifica que para esse atributo as cores se mantem sometne na faixa da cor azul. Como um exercício, veja os valores minimos e máximos para cada atributo e veja se o grafico corresponde corretamente a esses valores ao atribuir as cores.

Você pode escolher um limiar para atribuir uma determinada cor, e outro limiar atribuir uma outra cor bem diferente, para destacar, por exemplo, faixas de valroes de um determinado (ou determinados) atributo (s).

```
iris_scaled <- scale(iris %>% select(-Species))

ggplot(as_tibble(iris_scaled) %>% mutate(id = row_number()) %>% pivot_longer(cols = 1:4),
  aes(x = name, y = id, fill = value)) + geom_tile() +
  scale_fill_gradient2()
```



Correlation MAtrix

Esse tipo de gráfico permite calcular e visualizar as correlações entre os atributos.

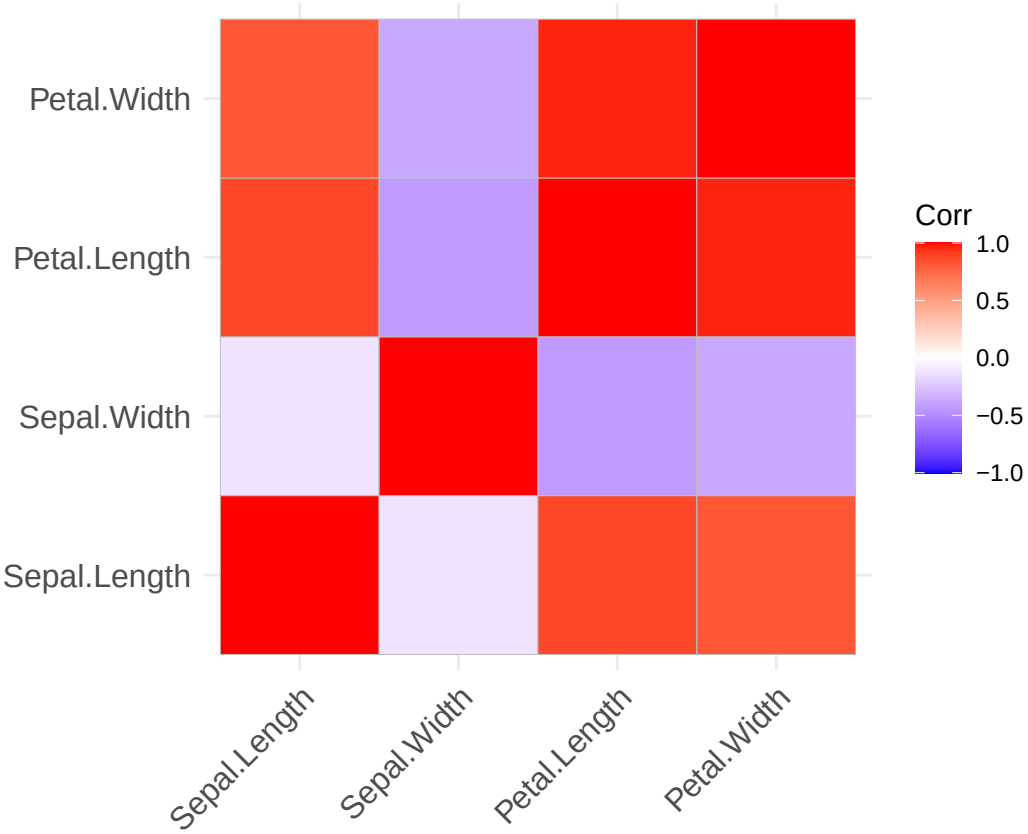
Primeiro calcula todas as correlações usando o comando "cor()". O atributo Species (categórico) não é considerado pois não é possível calculcar correlação de seus valores com os outros atributos.

```
cm1 <- iris %>% select(-Species) %>% as.matrix %>% cor()
cm1
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

Plotando as correlações:

```
library(ggcorrplot)
ggcorrplot(cm1)
```



Analisando esse gráfico, não considerando as correlaões dos atributos com eles mesmos (igual a 1, máxima), podemos ver pelas cores que as variáveis Petal.Width e Petal.Length são altamente correlacionadas (laranja quase vermelho).