Iris Dataset

1 Introduction

The iris dataset was introduced by the statistician Ronald Fisher and is one of the most frequently used datasets. Originally it was designed for linear discriminant analysis.

The set is a typical test case for many statistical classification techniques and has its own **wikipedia page.**

Figure 1: Types of iris flowers. Source: https://rpubs.com/vidhividhi/irisdataeda

We use OpenML (R-Package) to download the dataset in a machine-readable format and convert it into a `data.frame`:

```
# load the dataset from OpenML Library
d <- OpenML::getOMLDataSet(data.id = 61)

# convert the OpenML object to a tibble (enhanced data.frame)
iris <- d %>%
        dplyr::as_tibble() %>%
        dplyr::mutate(class = as.factor(stringr::str_extract(class, "[^-]+$")))
skimmed_iris <- skimr::skim(iris)
print(iris)
```

```
## # A tibble: 150 x 5
##    sepallength sepalwidth petallength petalwidth class
##          <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
```

The dataset has 3 classes, each with 50 instances, and each class represents a different species of iris plant. The dataset contains five columns: Sepal Length (`sepallength`), Sepal Width (`sepalwidth`), Petal Length (`petallength`), Petal Width (`petalwidth`), and the species type (`class`, the target of the dataset). The sepal length / width and petal length / width are recorded in [cm].

2 Exploratory Data Analysis (EDA)

In this part, we will walk through a few characteristics of iris dataset using library `skimr` and `DataExplorer`.

2.1 Factor variables

General statistics about factor variables from iris:

```
skimr::partition(skimmed_iris)$factor %>%
        knitr::kable(format = 'latex', booktabs = TRUE) %>%
        kableExtra::kable_styling(latex_options = 'HOLD_position')
```

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
class	0	1	FALSE	3	set: 50, ver: 50, vir: 50

There is only one factor variable in this dataset, and it is the iris class. There is no missing data and there are 3 values corresponding to 3 iris species types. The 150 data points are evenly distributed to 3 classes.

2.2 Numeric variables

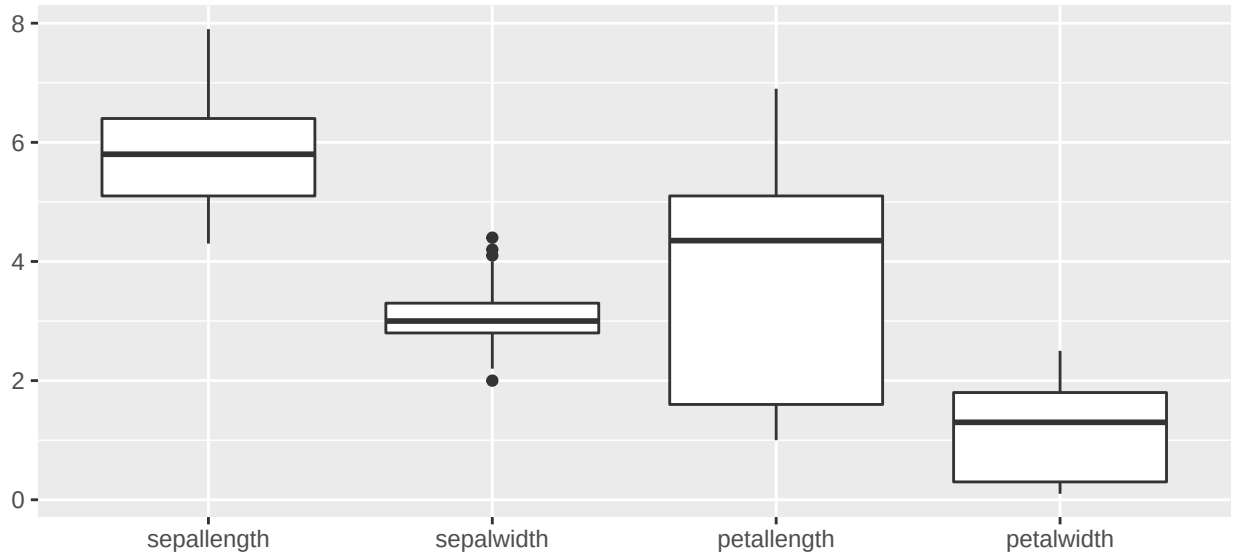
General statistics about numerical variables from iris:

```
skimr::partition(skimmed_iris)$numeric %>%
        knitr::kable(format = 'latex', booktabs = TRUE) %>%
        kableExtra::kable_styling(latex_options = 'HOLD_position')
```

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
sepallength	0	1	5.843333	0.8280661	4.3	5.1	5.80	6.4	7.9	
sepalwidth	0	1	3.054000	0.4335943	2.0	2.8	3.00	3.3	4.4	
petallength	0	1	3.758667	1.7644204	1.0	1.6	4.35	5.1	6.9	
petalwidth	0	1	1.198667	0.7631607	0.1	0.3	1.30	1.8	2.5	

Boxplots of numerical variables:

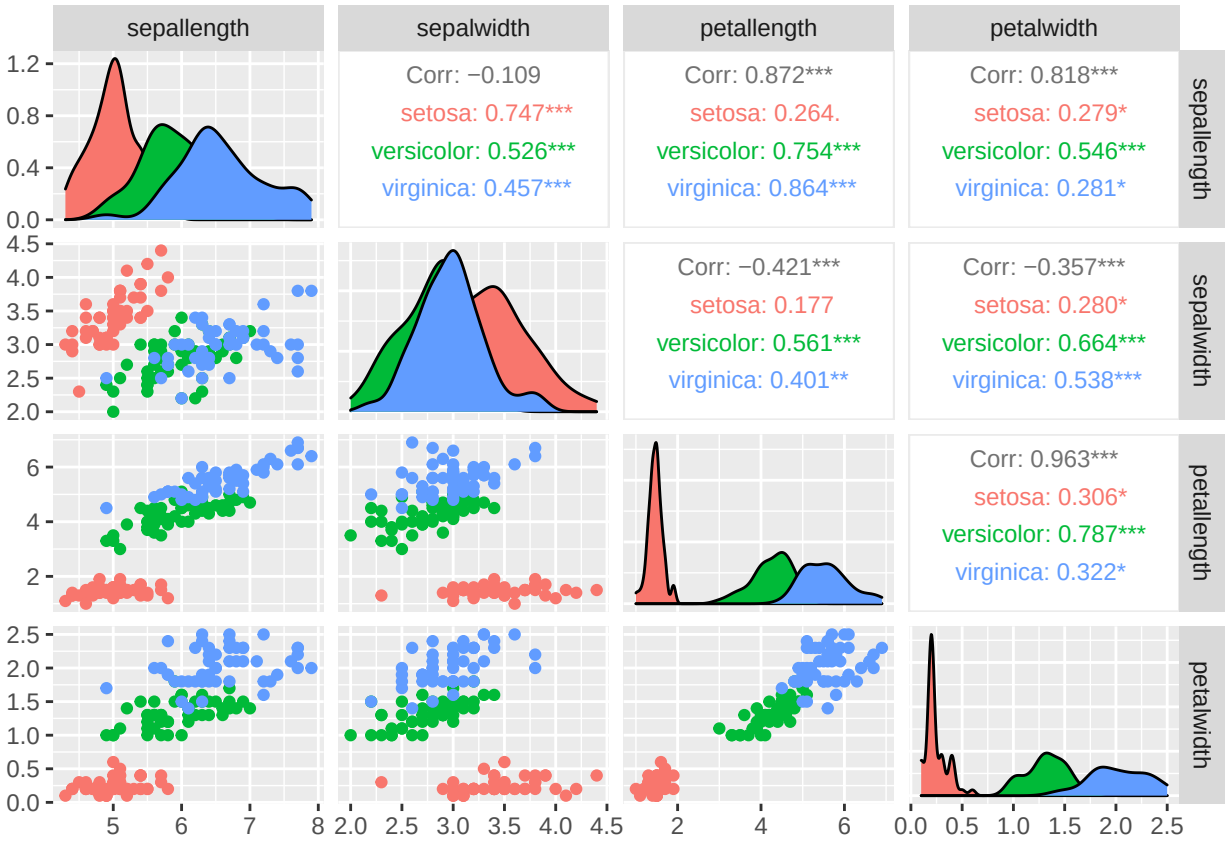
```
iris_numeric <- iris %>% select(where(is.numeric))
iris_numeric_melt <- melt(iris_numeric)
ggplot(data = iris_numeric_melt, aes(x=variable, y=value)) +
        geom_boxplot() +
        theme(axis.title.x=element_blank(),
              axis.title.y=element_blank())
```



Similar to the factor variable, numerical variables in this dataset don't have missing values and have only a few outliers with feature `sepalwidth`. From the statistics and the boxplots of the dataset's features, it can be seen that iris sepal length/width's size is generally larger than iris petal length/width's size.

Next, we will create a pairs plot using library `GGally` to check the correlations, the scatter plots of pairwise features and the distributions of classes for each feature.

```
GGally::ggpairs(iris,
                columns = 1:4,
                aes(color = class),
                upper = list(continuous = GGally::wrap("cor", size = 3))) +
        theme(text = element_text(size = 11))
```



From the plot above, looking at the diagonal plots, which are the density plots for classes in each feature, it can be seen that with only information from `petallength` or `petalwidth` alone, we can already easily distinguish `setosa` species from the other two species. To be more specific, it can be interpreted from the dataset that small `petallength` (< 2cm) or small `petalwidth` (< 0.5cm) is a great hint to predict the class to be `setosa`.

Regarding the correlation, it is worth noting that we can observe the *Simpson's Paradox* from some pairs of features such as `petallength-sepalwidth` and `petalwidth-sepalwidth`, i.e. for each class, those pairs are positively correlated but when the classes combine, this trend reverses.

About the features' pairwise scatter plots, it can be easily seen that the class setosa is linearly separable from the other two classes. The plots also indicate that different pairs of features can bring back much different level of separation between classes. For example, with the pair `sepallength-sepalwidth`, it is almost impossible to separate linearly the class `versicolor` and the class `virginica`, the opposite case is the pair `petallength-petalwidth`. This is a clear illustration about the importance of feature engineering and data analysis.