

Machine Learning - 1100-MLOENG (Ćwiczenia informatyczne Z-23/24)

[Home](#) > [My courses](#) > [Machine Learning - 1100-MLOENG \(Ćwiczenia informatyczne Z-23/24\)](#) > [Tidyverse](#) > [ggplot2](#)

ggplot2

The **ggplot2 package** is based on the principles of “The Grammar of Graphics” - hence “gg” in the name of ggplot2.

```
install.packages("ggplot2")  
library(ggplot2)
```

The main idea is to design a graphic as a succession of layers.

Every ggplot2 plot has the following key components:

1. the **dataset** that contains the variables that we want to represent. This is done with the **ggplot()** function and comes first.
2. the variable(s) to represent on the x and/or y-axis, and the **aesthetic elements** (such as color, size, fill, shape and transparency) of the objects to be represented. This is done with the **aes()** function.
3. the type of **graphical representation** - scatter plot, histogram, boxplot, etc.. This is done with the functions **geom_point()**, **geom_histogram()**, **geom_boxplot()**, etc.
4. if needed, additional layers (such as labels, annotations, scales, axis ticks, legends, themes, facets, etc.) can be added to personalize the plot.

Specifying the data

```
ggplot(house)
```

Then we add the variables to be represented with the **aes()** function

```
ggplot(house)+  
  aes(SqFt,Price)
```

The type of plot:

```
ggplot(house)+  
  aes(SqFt,Price)+  
  geom_point()
```

Data and aesthetic mappings are supplied in `ggplot()`, then layers are added on with `+`. This is an important pattern, and as you learn more about ggplot2 you'll construct increasingly sophisticated plots by adding on more types of components.

Titles and Axis Labels

You can add your own title and axis labels easily by incorporating following functions.

- **ggtitle()** - main plot title
- `xlab` - x-axis label
- `ylab` - y-axis label

```
ggplot(house)+  
  aes(SqFt,Price)+  
  geom_point()+  
  ggtitle("Houses") +  
  xlab("Area") +  
  ylab("Price ($)")
```

Change the Shape and Size of the Points

There are a very large number of options to improve the quality of the plot or to add additional information. These include:

- shape,
- symbol,
- size,
- color,
- alpha (transparency).

We can for instance change the shape of all points in a scatter plot by adding **shape** to **geom_point()**, or vary the shape according to the values taken by another variable (in that case, the **shape** must be inside **aes()**)

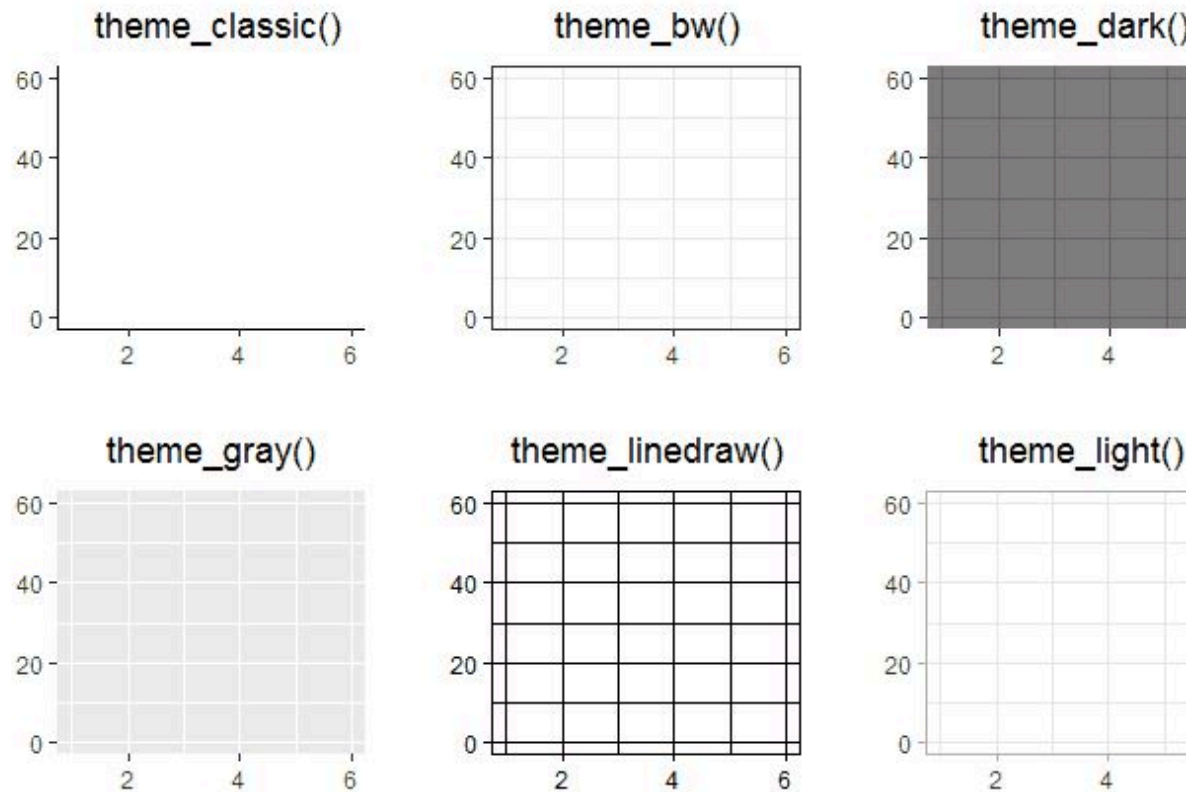
Symbols



Themes

The ggplot2 package provides some premade themes to change the overall plot appearance.

With themes you can easily customize some commonly used properties, like background color, panel background color and grid lines.



```
ggplot(house)+
  aes(SqFt,Price)+
  geom_point(shape="t", size=6.5, color="blue")+
  theme_classic()+
  ggtitle("Houses") +
  xlab("Area") +
  ylab("Price ($)")
```

Factor as color and shape

```
ggplot(house)+
  aes(SqFt,Price, colour=Neighborhood)+
  geom_point(shape=16, size=1.5)+
  theme_dark()
```

```
ggplot(house)+
  aes(SqFt,Price, shape= Brick, colour=Neighborhood)+
  geom_point( size=5.5)+
  theme_dark()
```

By default, the **legend** is located to the right side of the plot (when there is a legend to be displayed of course). To control the position of the legend, we need to use the `theme()` function in addition to the `legend.position` argument:

```
ggplot(house)+
  aes(SqFt,Price, shape= Brick, colour=Neighborhood)+
  geom_point( size=5.5)+
  theme(legend.position = "top")
```

Continuous Variable to Color or Size

```
house$sqft1=house$Price/house$SqFt
```

```
ggplot(house)+
  aes(sqft1,Price, colour= SqFt, shape=Brick, size=Bedrooms )+
  geom_point(alpha=0.3 )+
  theme_classic()
```

Plot geoms?

Examples of graphical representation functions

- [geom_boxplot\(\)](#) produces a box-and-whisker plot to summarise the distribution of a set of points.
- [geom_histogram\(\)](#) show the distribution of continuous variables.
- [geom_bar\(\)](#) shows the distribution of categorical variables.
- [geom_path\(\)](#) and [geom_line\(\)](#) draw lines between the data points. A line plot is constrained to produce lines that travel from left to right, while paths can go in any direction. Lines are typically used to explore how things change over time.

Last modified: czwartek, 27 października 2022, 6:51

Accessibility settings

Przetwarzanie danych osobowych

Platformą administruje Komisja ds. Doskonalenia Dydaktyki wraz z Centrum Informatyki Uniwersytetu Łódzkiego [Więcej](#)

Informacje na temat logowania

Na platformie jest wykorzystywana metoda logowania za pośrednictwem [Centralnego Systemu Logowania](#).

Studentów i pracowników Uniwersytetu Łódzkiego obowiązuje

Deklaracja dostępności

nazwa użytkownika i hasło
wykorzystywane podczas logowania
się do systemu USOSweb.