GGPlot

Marco Torchiano

Version 1.0.0 - April 2021

License 

Introduction

What is ggplot2?

The `ggplot2` is packagefor producing statistical graphics.

- ggplot2 is based on a grammar
  - allows composing graphs as combination of independent components
- ggplot2 takes care of fiddly details
  - defaults let produce publication-quality graphics in seconds
- ggplot2 is designed to work incrementally
  - start raw data, then add layers of annotations and statistical summaries

Graphics Grammar

Plot is composed of:

- data the information to be visualized (data frame)
- mapping of data onto aesthetic attributes
    - layer
        - geometric elements (geom)
        - statistical transformations (stat)
    - scale: maps data to attributes (e.g., color, size ..)
    - coord system: maps data coordinates to the plane
    - facet: breaks up the plot as small multiples
    - theme: provide support elements and controls details

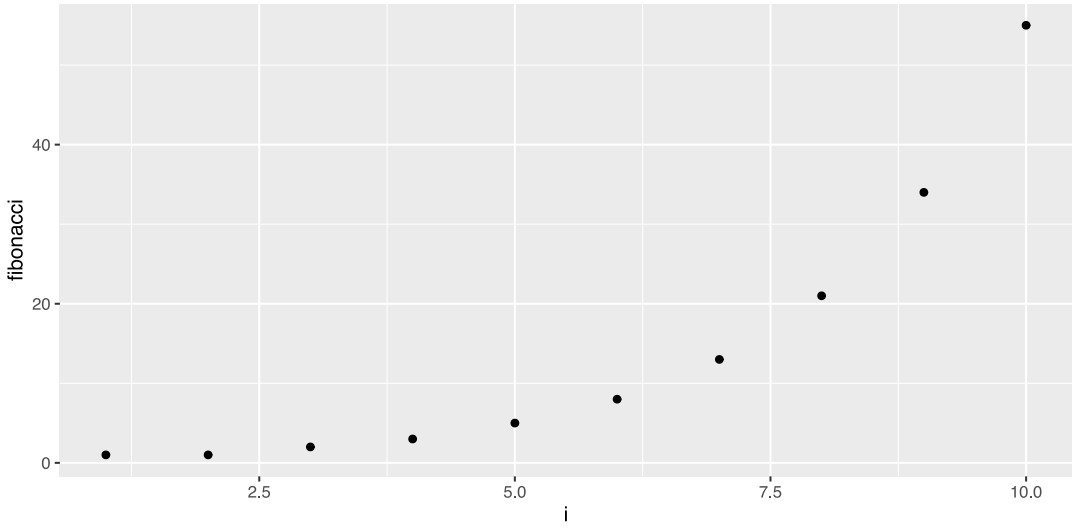
Basic elements

Any ggplot2 plot has three key components:

- the data
- aesthetic mappings
    - maps data variables to aesthetics features
    - coordinates or attributes
- visual layer (at least one)
    - define the visual object
    - maps aesthetics features to geometric properties

Basic elements

```
ggplot(series, aes(x=i,y=fibonacci))+geom_point()
```

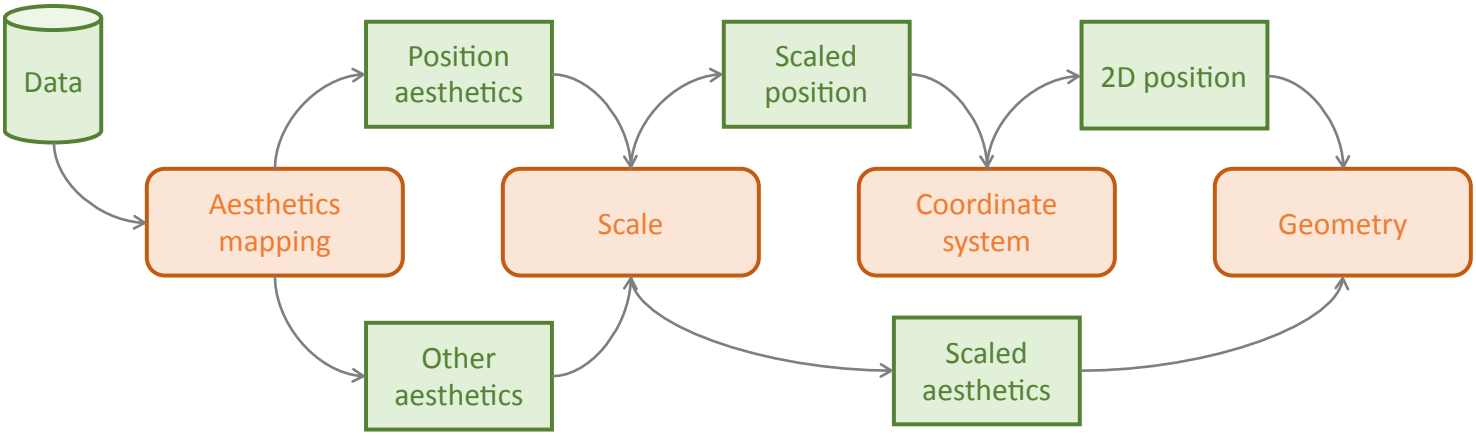


Basic elements

```
ggplot(series, aes(x=i,y=fibonacci))+geom_point()
```

- `series` : defines the data to be used
- `aes(x=i,y=fibonacci)` : maps data to visual characteristics
  - `i` and `fibonacci` to the x and y coordinates respectively
  - cartesian coordinates are implied by default
  - linear scales implied
- `geom_point()` : defines a layer that map data to points
  - shape, color, size of points are implied by default

Mappings



- Scale depends on the type of aesthetics
  - for position (x, y) is by default a simple linear scale
  - for other types of aesthetics may vary

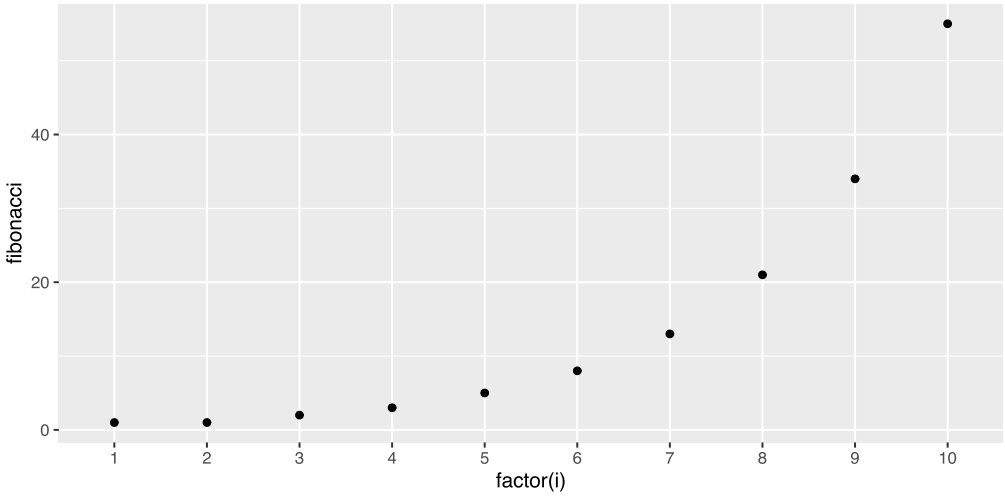
Scales and coordinates

Both scale and coordinates have (implicit) defaults:

- the default scale depends on
  - the specific aestethics
  - the type of the variable
- the default coordinate system is `coord_cartesian()`
  - another option is `coor_polar()`

Default scale adapts to variable

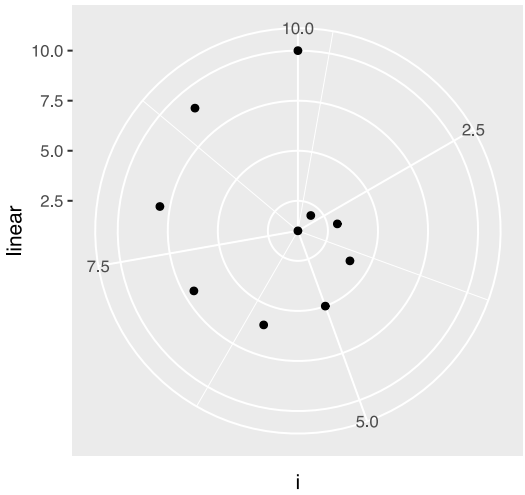
```
ggplot(series, aes(x=factor(i),y=fibonacci))+geom_point()
```



A `factor` is mapped to equidistanced slots along the axis

Different coordinate system

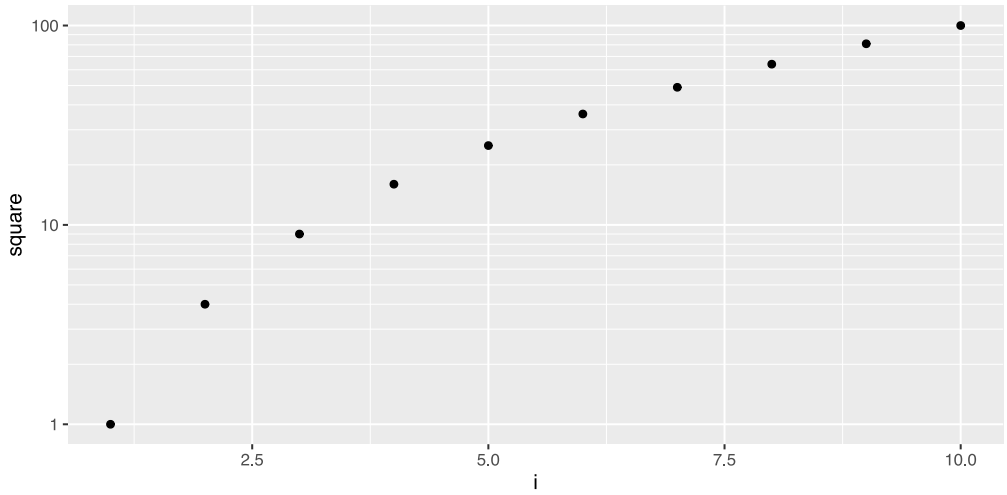
```
ggplot(series, aes(x=i,y=linear))+geom_point()+
      coord_polar()
```



`x` maps to θ (with `max(x)` $\rightarrow 2\pi$) and `y` maps to ρ

Different y axis scale

```
ggplot(series, aes(x=i,y=square))+geom_point()+
      scale_y_log10(minor_breaks=c(1:10,1:10*10))
```



Applied a log scale to the position `y`

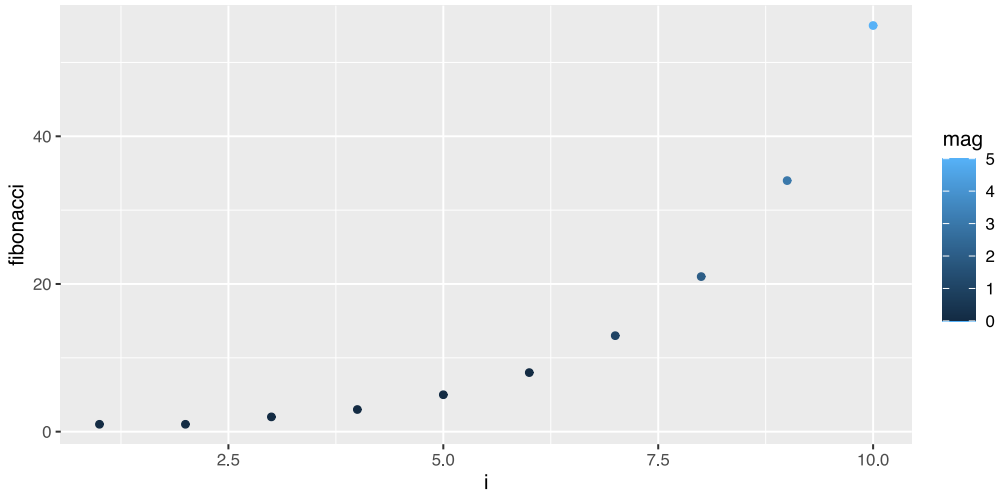
Additional aesthetics

Aesthetics include:

- position (`x`, `y`)
- grouping (`group`)
- other:
  - `color` : line or simbol color
  - `fill` : area fill color
  - `shape` : type of shape
  - `size` : size of the object

Additional aesthetics

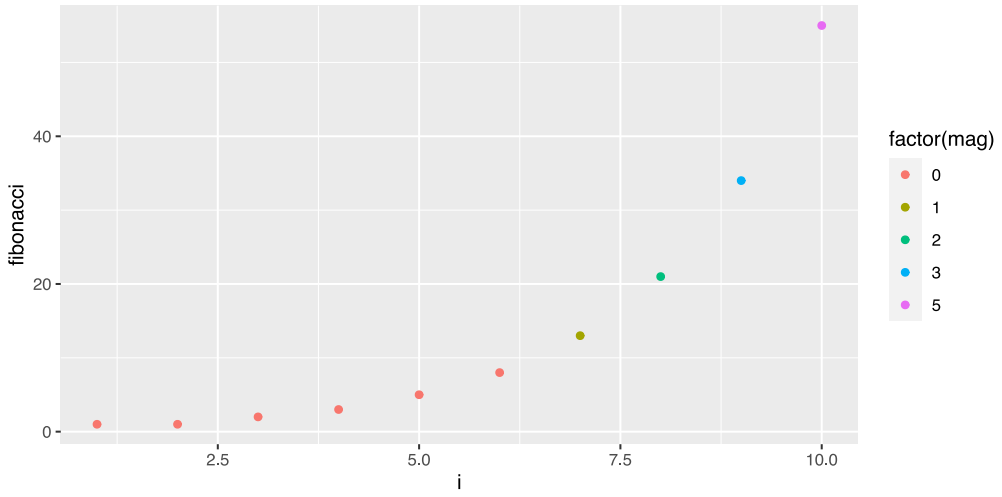
```
ggplot(series %>% mutate( mag = fibonacci %/% 10),
       aes(x=i, y=fibonacci, color=mag))+ geom_point()
```



A gradient scale is used for a continuous (`numeric`) variable

Additional features

```
ggplot(series%>%mutate( mag = fibonacci %/% 10),
       aes(x=i, y=fibonacci, color=factor(mag)))+ geom_poi
```



Discrete color scale is used for a `factor` variable

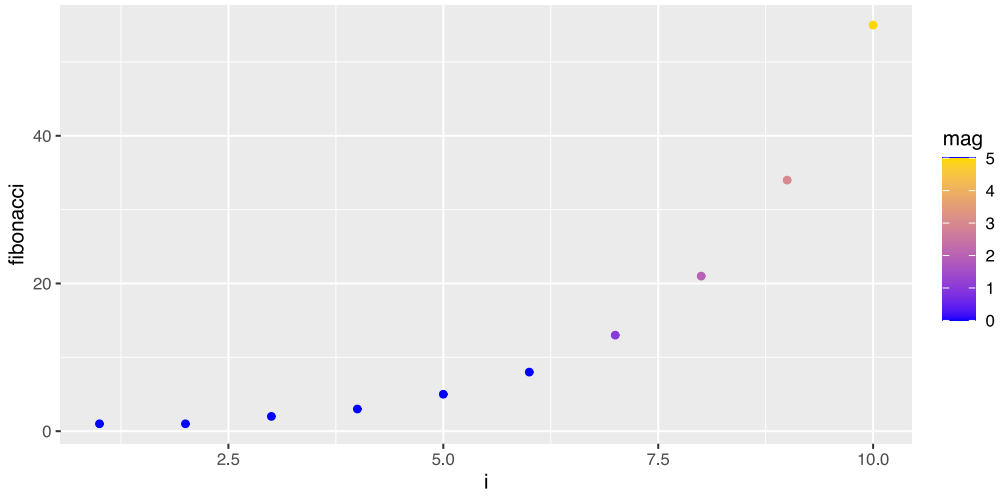
Scales

For each aesthetics type a few scales are provided:

- `scale_x_..`, `scale_y_..`
- `scale_color_..`
- `scale_fill_..`
- `scale_shape_..`
- `scale_size_..`

Additional feature and scale

```
ggplot(series%>%mutate( mag = fibonacci %/% 10),
       aes(x=i, y=fibonacci, color=mag))+
       scale_color_gradient(low="blue",high="gold")+
       geom_point()
```



Geometry layers

Geometry function add new layers

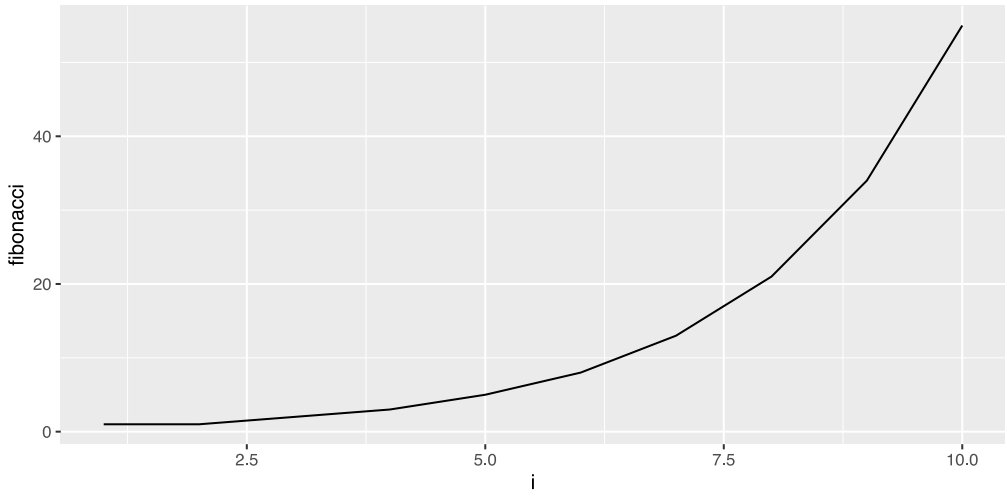
- `geom_point()` : draw points
- `geom_line()` : draw lines connecting positions
- `geom_text()` and `geom_label()` : write a text or label
- `geom_area()` : draw a filled area

Layers are drawn in order of declaration, with the latest on top.

The order of all other statements is irrelevant.

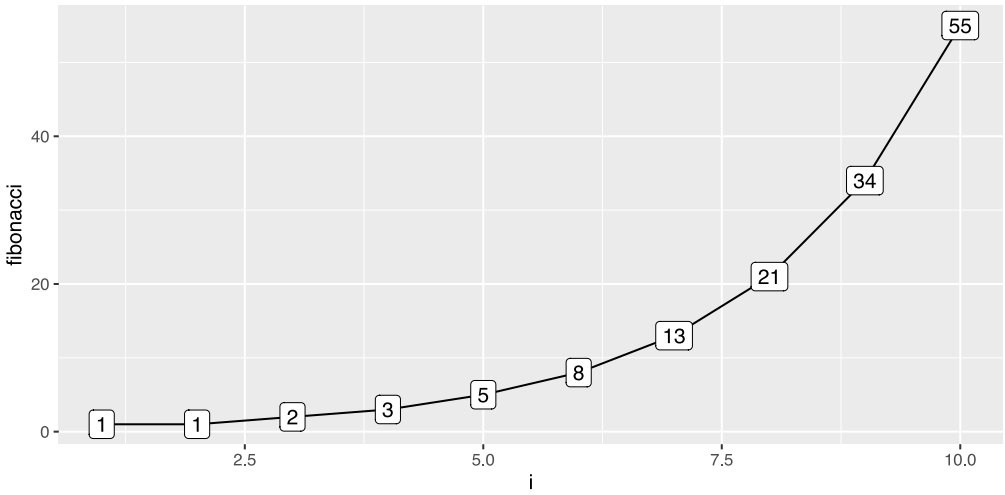
Changing geometry

```
ggplot(series, aes(x=i, y=fibonacci))+
       geom_line()
```



Using multiple layers

```
ggplot(series, aes(x=i, y=fibonacci, label=fibonacci))+
        geom_line() + geom_label()
```



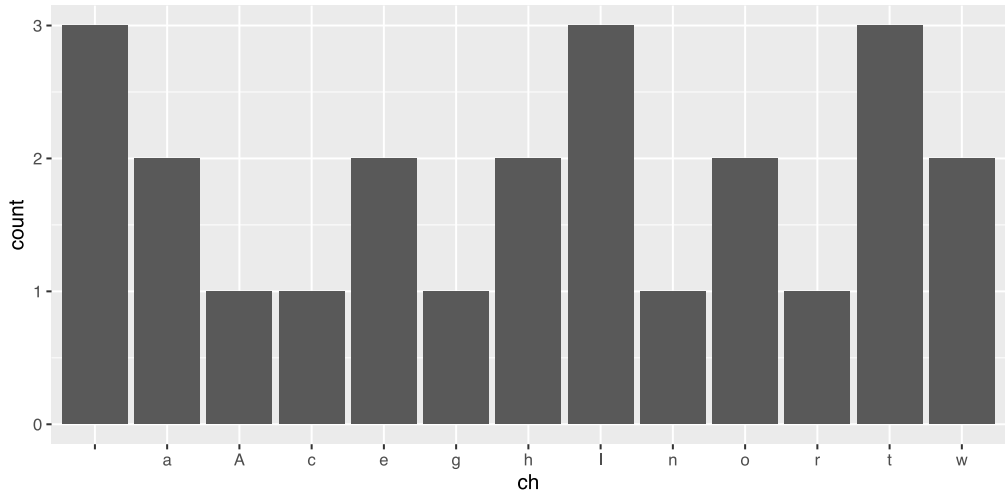
Geometries with statistical transformation

A few geometries perform a transformation befor mapping to an object

- `geom_bar()` : compute frequencies of discrete variables
- `geom_histogram()` : compute frequencies of bins of continuous vars
- `geom_boxplot()` : compute boxplot
- `geom_violin()`: compute a violin plot

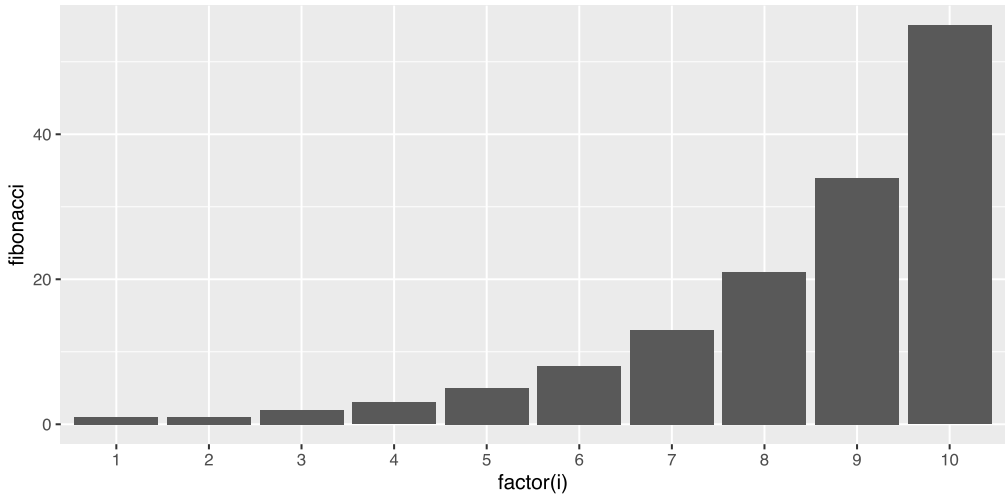
Computing frequencies

```
ch = strsplit("All along the watchtower",c())[[1]]
ggplot(data.frame(ch=ch), aes(x=ch))+ geom_bar()
```



Regular barplot

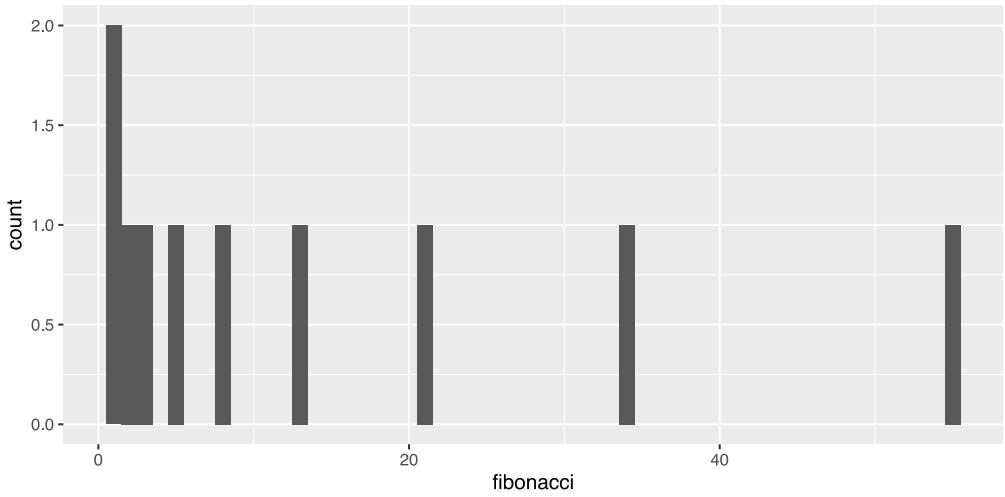
```
ggplot(series, aes(x=factor(i),y=fibonacci))+
      geom_bar(stat="identity")
```



Conventional bar plot uses stat `identity` (instead of `count`)

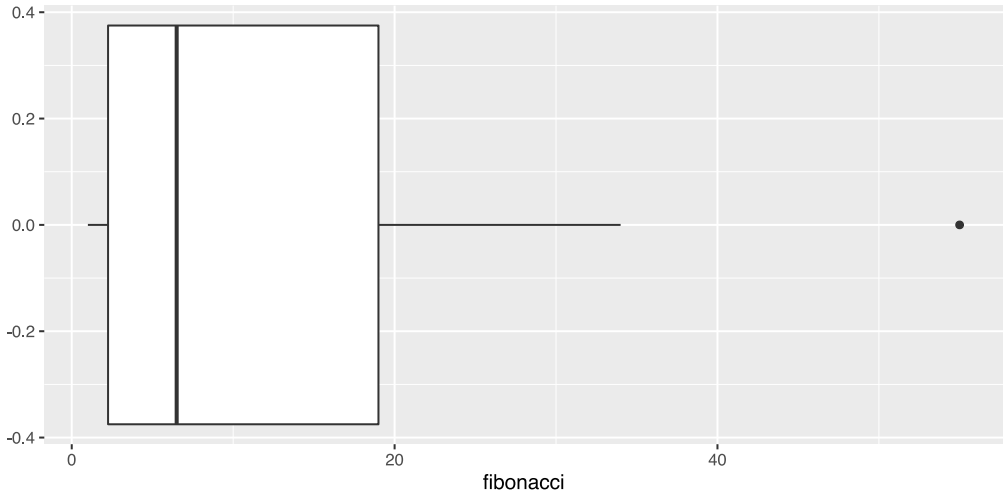
Histogram geometry

```
ggplot(series, aes(x=fibonacci))+
      geom_histogram(binwidth=1)
```



Boxplot geometry

```
ggplot(series, aes(x=fibonacci))+
      geom_boxplot()
```



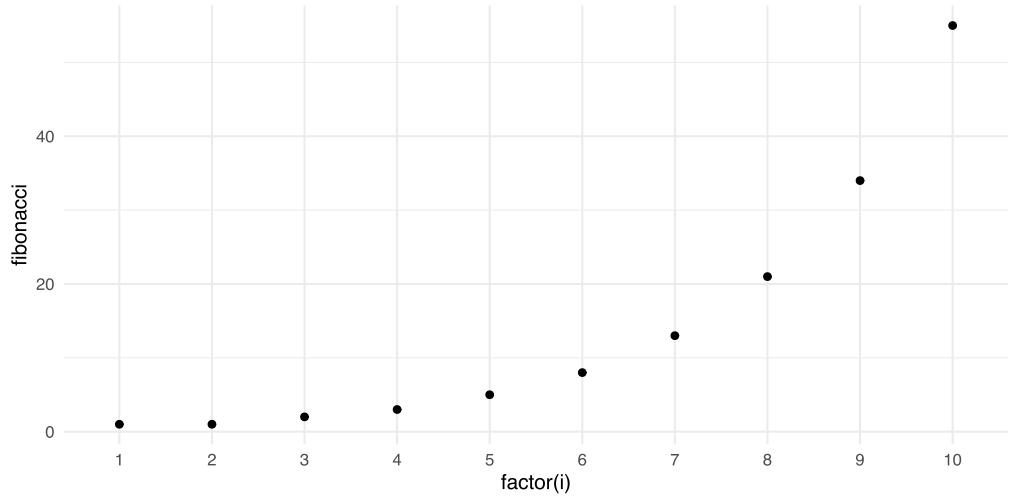
Theme

The support elements and default visual features are defined by a theme

- `theme_classic()` : similar to base functions
- `theme_gray()` : the default theme (gray background)
- `theme_bw()` : same as default but with white backgound
- `theme_light()` : same as bw but with lighter lines
- `theme_dark()` : dark gray background
- `theme_minimal()` : minimalistic theme
- `theme_void()` : no supporting elements

Changing the theme

```
ggplot(series, aes(x=factor(i),y=fibonacci))+geom_point()+
    theme_minimal()
```



References

- Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen. “ggplot2: Elegant Graphics for Data Analysis”, in-prograss
    - https://ggplot2-book.org/
- Winston Chang, “R Graphics Cookbook” O’Reilly, 2013