Now it's time to use customed fonts!!!

耿远昊

March 2020

Formula:

First, we will test on LaTeX: $\Sigma_{k=1}^{n} k^2$, $\iint_{\mathbb{R}^2} \dfrac{x}{y} d\omega$. It seems perfect!
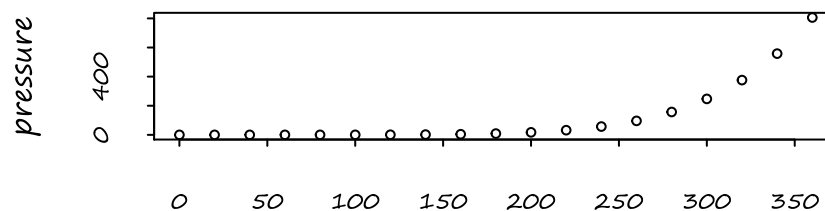
$$\sum_{n=1}^{\infty} 2^{-n} = 1$$

Plots:

Then, let us check several key points:

(a) r-base plot

```
library(showtext)
showtext_auto(enable = TRUE)
font_add('SegoePrint', 'SegoePrint.ttf')
plot(pressure,xlab='You can change the fonlubridatet now!',family='SegoePrint',tck=-0.05
                                            ,main='r-base plot'
                                            ,cex=0.6,cex.main=1
                                            ,cex.lab=0.9,cex.axis=0.7)
```


r-base plot

You can change the fonlubridatet now!

(b) ggplot2

```
library(ggplot2)
library(quantmod)
data("economics", package = "ggplot2")
```

```
economics$returns_perc <- c(0, diff(economics$psavert)/
economics$psavert[-length(economics$psavert)])
brks <- economics$date[seq(1, length(economics$date), 12)]
lbls <- lubridate::year(economics$date[seq(1,
length(economics$date), 12)])
ggplot(economics[1:100, ], aes(date, returns_perc)) +
    geom_area() +
    scale_x_date(breaks=brks, labels=lbls) +
    theme(axis.text.x = element_text(angle=90)) +
    labs(title="Area Chart",
    y="Returns for Personal savings",
    caption="Source: economics")+
    theme(plot.title = element_text(hjust = 0.5))+
    theme(text=element_text(family="SegoePrint",size=9))
```


Area Chart
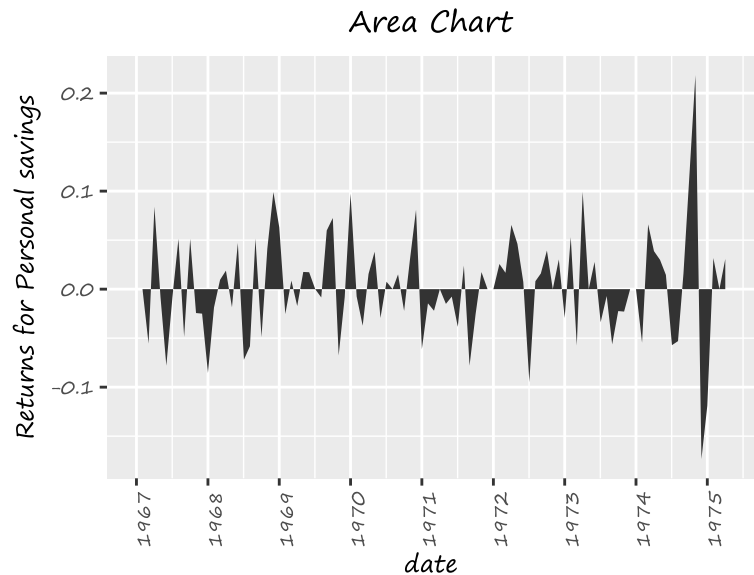
Source: economics
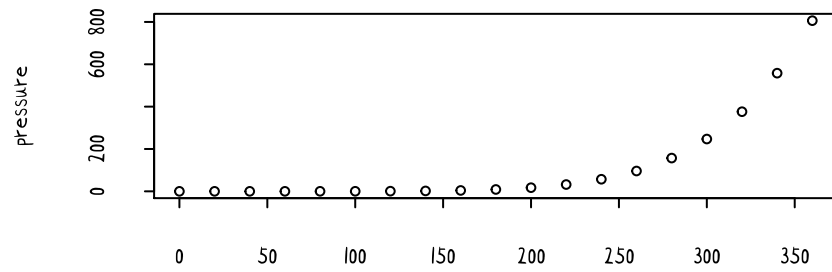
(c) r-base plot with chinese

```
font_add('Seafood', 'Seafood.ttf')
plot(pressure,xlab=' 现在可以更改字体了！',family='Seafood',tck=-0.05
                                        ,main=' 基础包作图'
                                        ,cex=0.6,cex.main=1.3
                                        ,cex.lab=1.2,cex.axis=1.05)
```
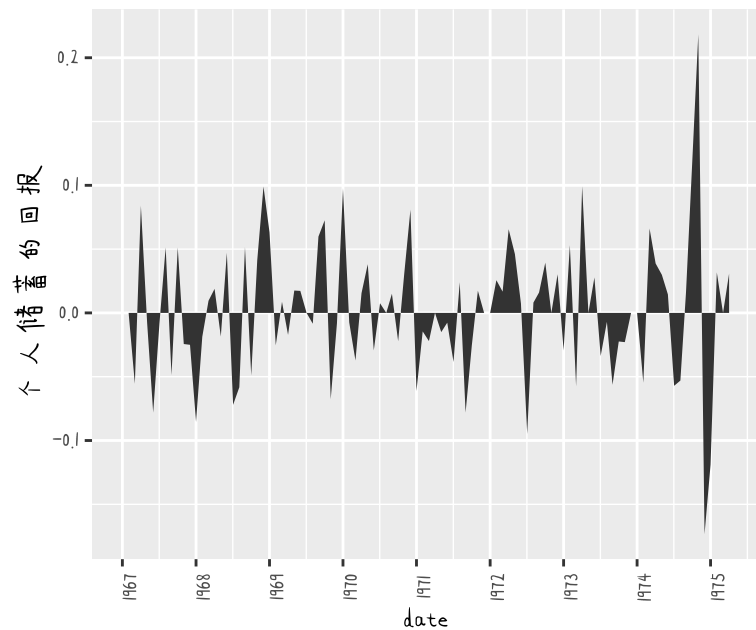
基础包作图

现在可以更改字体了!

(d) ggplot2 with chinese

```r
library(ggplot2)
library(quantmod)
ggplot(economics[1:100, ], aes(date, returns_perc)) +
    geom_area() +
    scale_x_date(breaks=brks, labels=lbls) +
    theme(axis.text.x = element_text(angle=90)) +
    labs(title="分区图",
    y="个人储蓄的回报",
    caption="来源: Economics")+
    theme(plot.title = element_text(hjust = 0.5))+
    theme(text=element_text(family="Seafood",size=13))
```

分区图

来源: Economics

Table:

First name	Last name	Grade
John	B	7.5
Bob	A	5.2
Tom	E	6.4
Gary	D	2.3
Richard	A	2
Bach	C	0.6
6 Person	A	7.5

Algorithm

Algorithm 1 AdaBoost 二分类问题

Input: M 个弱二分类器，N 个训练样本，迭代轮数 T

Output: H_{Boost}

1: $\alpha_0 \leftarrow 0$

2: $h_0(x_i) \leftarrow 0, i = 1, ..., N$

3: **for each** $t \in [1, T]$ **do**

4:    $H_{t-1} \leftarrow \sum_{i=0}^{t-1} \alpha_i h_0(x_i)$

5:    $w_{t,i} \leftarrow e^{-y_i H_{t-1}(x_i)}$

6:    $h_t \leftarrow \underset{h}{\arg\min} \sum_{i=1}^{N} w_{t,i} \mathbb{I}(y_i = h(x_i))$

7:    $W_{t,i} \leftarrow \dfrac{w_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$

8:    $e_t \leftarrow \sum_{i=1}^{N} W_{t,i} \mathbb{I}(y_i \neq h_t(x_i))$

9:    **if** $e_t > \dfrac{1}{2}$ **then**

10:      Break

11:    **else**

12:      $\alpha_t \leftarrow \dfrac{1}{2} \ln \dfrac{1 - e_t}{e_t}$

13:    **end if**

14: **end for**

15: **return** $H_{Boost} = sign(\sum_{i=1}^{T} \alpha_i h_i(x))$

How to do it?

(1) You should find your system font folder.

(2) Download the *.ttf file which represent your customed font.

(3) Copy the file into your font folder.

(4) Download `showtext` package.

(5) Use `showtext_auto(enable = TRUE)` and `font_add('*', '*SegoePrint.ttf*.ttf')` to register your font.

(6) Change the font family setting in your code.

(7) Change the chunk head with `fig.showtext=TRUE`, or it won't work!

(8) Add head setting in your Rmd file. In this example, it was wirtten like this:

```
output:
  pdf_document:
    latex_engine: xelatex
  word_document: default
header-includes:
  - \usepackage{ctex}
  - \usepackage{fontspec}
  - \setmainfont{SegoePrint}
  - \everymath{\displaystyle}
  - \usepackage{amsmath}
  - \usepackage{booktabs}
  - \usepackage{algorithm}
  - \usepackage{algorithmic}
  - \usepackage{graphicx}
  - \usepackage{amssymb}
  - \usepackage{amsfonts}
  - \DeclareMathOperator*{\argmax}{argmax}
  - \DeclareMathOperator*{\argmin}{argmin}
  - \renewcommand{\algorithmicrequire}{ \textbf{Input:}}
  - \renewcommand{\algorithmicensure}{ \textbf{Output:}}
```

图 1: Rmd Head of This Example

If you have the same problem with matplotlib in python, you can look this instruction about how to configurate default custom fonts in matplotlib. At least, it works for me after looking into a large number of referances (which some of them make me painful and confused).

Hope this guide can help you!